

```

UUU      UUU      EEEEEEEEEEEEEEE      TTTTTTTTTTTTTTTT      PPPPPPPPPPPP      SSSSSSSSSSSSS      YYY      YYY
UUU      UUU      EEEEEEEEEEEEEEEEE      TTTTTTTTTTTTTTTT      PPPPPPPPPPPP      SSSSSSSSSSSSS      YYY      YYY
UUU      UUU      EEEEEEEEEEEEEEEEE      TTTTTTTTTTTTTTTT      PPΓPPPPPPPPP      SSSSSSSSSSSSS      YYY      YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEE                      TTT          PPP          PPP      SSS          YYY          YYY
UUU      UUU      EEEEEEEEEEEEEEE      TTT          PPPPPPPPPPPPP      SSSSSSSSSSS      YYY
UUU      UUU      EEEEEEEEEEEEEEEEE      TTT          PPPPPPPPPPPPP      SSSSSSSSSSS      YYY
UUU      UUU      EEEEEEEEEEEEEEE      TTT          PPPPPPPPPPPPP      SSSSSSSSSSS      YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUU      UUU      EEE                      TTT          PPP          SSS          YYY
UUUUUUUUUUUUUUUUUU      EEEEEEEEEEEEEEEEE      TTT          PPP          SSSSSSSSSSSSS      YYY
UUUUUUUUUUUUUUUUUU      EEEEEEEEEEEEEEEEE      TTT          PPP          SSSSSSSSSSSSS      YYY
UUUUUUUUUUUUUUUUUU      EEEEEEEEEEEEEEEEE      TTT          PPP          SSSSSSSSSSSSS      YYY

```

[illegible]

[illegible][illegible]

(1)	77	DECLARATIONS
(1)	332	R/W PSECT
(1)	413	SATSSS35
(1)	462	CREPRC TESTS
(1)	764	GETJPI TESTS
(2)	969	ROUTINES
(2)	970	REG_SAVE
(2)	991	REG_CHECK
(2)	1033	PRINT_FAIL
(2)	1080	MODE_ID
(2)	1102	CRE_CHECK
(2)	1143	JPI_CHECK

```
0000 1 .TITLE SATSSS35 - SATS SYSTEM SERVICE TESTS (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: SATS SYSTEM SERVICE TESTS
0000 31
0000 32 ABSTRACT: The SATSSS35 module tests the execution of the following
0000 33 VMS system services:
0000 34
0000 35 $CREPRC
0000 36 $GETJPI
0000 37
0000 38 ENVIRONMENT: User mode image.
0000 39 Needs CMKRN privilege and dynamically acquires other
0000 40 privileges, as needed.
0000 41
0000 42 AUTHOR: Larry D. Jones, CREATION DATE: JULY, 1979
0000 43
0000 44 MODIFIED BY:
0000 45
0000 46 V03-002 LDJ0006 Larry D. Jones, 23-Mar-1982
0000 47 Made the quota list be absolute minimum to test the
0000 48 SYSBOOT minimum values.
0000 49
0000 50 V03-001 RNP0005 Robert N. Perron, 23-Mar-1982
0000 51 Removed EXCVEC and FINALEXC from the JPI_GOOD list.
0000 52
0000 53 V02-006 RNP0004 Robert N. Perron, 09-Dec-1981
0000 54 Removed ASTEN from the JPI_GOOD list.
0000 55
0000 56 V02-005 RNP0003 Robert N. Perron, 02-Oct-1981
0000 57 Removed ASTACT from the JPI_GOOD list.
```

0000	58	:	
0000	59	:	
0000	60	:	V02-004 LDJ0002 Larry D. Jones, 06-Sep-1981
0000	61	:	Fixed GETJPI P1 reference to Ctl\$AQ_EXCVEC and Ctl\$AL_FINALEXC.
0000	62	:	
0000	63	:	V02-003 RNP0002 Robert N. Perron, 01-Jun-1981
0000	64	:	To eliminate dependence on the SYSTEST account privileges
0000	65	:	being a specific list, privileges are now set to a fixed list
0000	66	:	before the GETJPI tests are started.
0000	67	:	
0000	68	:	V02-002 RNP0001 Robert N. Perron, 09-Apr-1981
0000	69	:	Fixed problem of STS field changing due to Swapper activity.
0000	70	:	Prevent failure when privileges are added to the SYSTEST
0000	71	:	account. Cleaned up some format problems.
0000	72	:	
0000	73	:	V02-001 LDJ0001 Larry D. Jones, 17-Sep-1980
0000	74	:	Modified to conform to new build command procedures.
0000	75	:	

```
0000 77 .SBTTL DECLARATIONS
0000 78 :
0000 79 : MACRO LIBRARY CALLS
0000 80 :
0000 81 $ACCDDEF ; account record offset definitions
0000 82 $DIBDEF ; device info block definitions
0000 83 $JPIDEF ; JPI offset definitions
0000 84 $PCBDEF ; process control block definitions
0000 85 $PHDDEF ; Process header definitions
0000 86 $PQLDEF
0000 87 $PRVDEF ; privilege definitions
0000 88 $SHRDEF ; shared message definitions
0000 89 $SFDEF ; stack frame definitions
0000 90 $STSDEF ; STS definitions
0000 91 $UETPDEF ; UETP message definitions
0000 92 :
0000 93 :
00000001 0000 94 SUCCESS = 1 ; success
00000002 0000 95 ERROR = 2 ; error
0000 96 :
0000 97 : SHR message definitions
0000 98 :
00740000 0000 99 UETP = 116@STS$V_FAC_NO ;define the UETP facility code
0000 100 :
00741038 0000 101 UETP$_BEGIN = UETP!SHR$_BEGIN ;define the UETP messages
00741130 0000 102 UETP$_TEXT = UETP!SHR$_TEXT
007410E0 0000 103 UETP$_ABENDD = UETP!SHR$_ABENDD
00741080 0000 104 UETP$_ENDEDD = UETP!SHR$_ENDEDD
0000 105 :
0000 106 : Mask of bits for the STS field in a $CREPRC system service as they are
0000 107 : returned from a $GETJPI system service.
0000 108 :
0000 109 JPI_STS_MASK = <<1@PCB$V_NETWORK>!<1@PCB$V_SSFEXCU>!<1@PCB$V_SSRWAIT>!-
0000 110 <1@PCB$V_BATCH>!<1@PCB$V_NOACNT>!<1@PCB$V_HIBER>!-
0038C600 0000 111 <1@PCB$V_LOGIN>>
0000 112 :
0000 113 : The opposite of JPI_STS_MASK
0000 114 :
FFC739FF 0000 115 JPI_STS_NMASK = ^CJPI_STS_MASK
0000 116 :
0000 117 :
0000 118 : Mask of bits for the Privilege field as they are returned from a $GETJPI
0000 119 :
0000 120 JPI_PRV_MASK = <<1@PRV$V_CMEXEC>!<1@PRV$V_CMKNL>!<1@PRV$V_DETACH>!-
0000 121 <1@PRV$V_DIAGNOSE>!<1@PRV$V_GROUP>!<1@PRV$V_GRPNAM>!-
0000 122 <1@PRV$V_LOG_IO>!<1@PRV$V_NETMBX>!<1@PRV$V_NOACNT>!-
0000 123 <1@PRV$V_PHY_IO>!<1@PRV$V_PRMCB>!<1@PRV$V_PRRMBX>!-
0000 124 <1@PRV$V_PSWAPM>!<1@PRV$V_SETPRI>!<1@PRV$V_SYSNAM>!-
1070BF EF 0000 125 <1@PRV$V_SYSPRV>!<1@PRV$V_TPMBX>!<1@PRV$V_VOLPRO>>
0000 126 :
0000 127 : The compliment of JPI_PRV_MASK
0000 128 :
EF8F4010 0000 129 JPI_PRV_NMASK = ^CJPI_PRV_MASK
0000 130 :
0000 131 : MACROS
0000 132 :
0000 133 .MACRO JPI,NAME,SIZE
```

SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00 Page 4
DECLARATIONS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1 (1)

```
0000 134 .WORD SIZE
0000 135 .WORD JPI$ 'NAME'
0000 136 .ADDRESS NAME
0000 137 .ADDRESS NAME'L
0000 138 .SAVE PSECT
0000 139 .PSECT ITEM_LIST
0000 140 NAME:
0000 141 .BLKB SIZE
0000 142 NAME'L:
0000 143 .WORD 0
0000 144 .RESTORE PSECT
0000 145 .ENDM JPI
```

SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 5
(1)

```
00000000 147 .PSECT ITEM_LIST, RD, WRT, NOEXE, LONG ; psect to store JPI results in
00000000 148 .PSECT RODATA, RD, NOWRT, NOEXE, LONG
0000 149 ;
0000 150 TEST_MOD_NAME:
0000 151 .ASCIC /SATSSS35/ ; needed for SATSMS message
08 0000
0009 152 TEST_MOD_NAME_D:
0009 153 .ASCIC /SATSSS35/ ; module name
35 33 0017
0019 154 TEST_MOD_BEGIN: ; start end and fail messages
0019 155 .ASCIC /begun/
001F 156 TEST_MOD_SUCC:
001F 157 .ASCIC /successful/
0A 001F
002A 158 TEST_MOD_FAIL:
002A 159 .ASCIC /failed/
06 002A
0031 160 CS1: ; failure messages
0031 161 .ASCID \Test !AC service name !AC step !UL failed.\
003F
004B
0057
0063 162 CS2:
0063 163 .ASCID \Expected !AS = !XL received !AS = !XL\
0071
007D
0089
0090 164 CS3:
0090 165 .ASCID \Expected !AS!UB = !XL received !AS!UB = !XL\
009E
00AA
00B6
00C2
00C3 166 CS5:
00C3 167 .ASCID \Mode was !AS.\
00D1
00D8 168 EXP:
00D8 169 .ASCID \status\
00E6 170 AST_PARAM:
00E6 171 .ASCID \AST param.\
00F4
00F8 172 BP:
00F8 173 .ASCID \base pri.\
0106
0109 174 PNS:
0109 175 .ASCID \Process name was not set correctly.\
0117
0123
012F
0134 176 STSFLGS:
0134 177 .ASCID \STSFLG's\
0142
0144 178 UIC_MSG:
0144 179 .ASCID \UIC\
014F 180 EFC_NAME:
014F 181 .ASCID \SATSSF06_DET\
```

```
54 45 44 5F 36 30 015D
44 49 50 0000016B'010E0000' 0163 182 PID_STR:
43 52 50 45 52 43 00' 0163 183 .ASCID \PID\
06 016E 184 CREPRC:
016E 185 .ASCIC \CREPRC\
0175 186 GETJPI:
49 50 4A 54 45 47 00' 0175 187 .ASCIC \GETJPI\
06 0175
017C 188 UM: ; mode messages
72 65 73 75 00000184'010E0000' 017C 189 .ASCID \user\
0188 190 MBNAM:
58 42 4D 35 33 53 00000190'010E0000' 0188 191 .ASCID \S35MBX\
0196 192 PRVMASK:
00000000 1070BF EF 0196 193 .QUAD JPI_PRV_MASK ; used for setting privileges to
019E 194 ; known value
019E 195 NPRVMASK:
00000000 EF8F4010 019E 196 .QUAD JPI_PRV_NMASK ; used for clearing any extra
01A6 197 ; privileges
01A6 198 MSGVEC: ; PUTMSG message vector
00000003 01A6 199 .LONG 3
00741130 01AA 200 .LONG UETPS_TEXT
00000001 01AE 201 .LONG 1
00000217' 01B2 202 .ADDRESS MESSAGEL
01B6 203 QUOTA_LIST:
01 01B6 204 .BYTE PQLS_ASTLM ; minimum quota list
00000001 01B7 205 .LONG 1
02 01BB 206 .BYTE PQLS_BIOLM
00000001 01BC 207 .LONG 1
03 01C0 208 .BYTE PQLS_BYTLM
00000001 01C1 209 .LONG 1
04 01C5 210 .BYTE PQLS_CPULM
00000000 01C6 211 .LONG 0
05 01CA 212 .BYTE PQLS_DIOLM
00000001 01CB 213 .LONG 1
06 01CF 214 .BYTE PQLS_FILLM
00000001 01D0 215 .LONG 1
07 01D4 216 .BYTE PQLS_PGFLQUOTA
00000001 01D5 217 .LONG 1
08 01D9 218 .BYTE PQLS_PRCLM
00000000 01DA 219 .LONG 0
09 01DE 220 .BYTE PQLS_TQELM
00000000 01DF 221 .LONG 0
0B 01E3 222 .BYTE PQLS_WSDEFAULT
00000001 01E4 223 .LONG 1
0A 01E8 224 .BYTE PQLS_WSQUOTA
00000001 01E9 225 .LONG 1
00 01ED 226 .BYTE PQLS_LISTEND
```

```
01EE 228 GET_LIST: ; GETJPI list of items and results
01EE 229 JPI ACCOUNT,8
01FA 230 SHORT_LIST:
01FA 231 JPI CPULIM,4
0206 232 JPI CURPRIV,8
0212 233 JPI GRP,4
021E 234 JPI IMAGPRIV,8
022A 235 JPI MEM,4
0236 236 JPI PRCLM,4
0242 237 JPI TQLM,4
024E 238 JPI UIC,4
025A 239 JPI USERNAME,12
00000044 0266 240 JPI LIST_SIZE=<USERNAME+2>-ACCOUNT
0000003A 0266 241 JPI LIST_SIZE1=<USERNAME+2>-CPULIM
0266 242 DIRTY: ; GETJPI entrys which will vary
0266 243 JPI APTCNT,4
0272 244 JPI ASTACT,4
027E 245 JPI ASTEN,4
028A 246 JPI ASTCNT,4
0296 247 JPI ASTLM,4
02A2 248 JPI AUTHPRIV,8
02AE 249 JPI BIOCNT,4
02BA 250 JPI BIOLM,4
02C6 251 JPI BUFIO,4
02D2 252 JPI BYTCNT,4
02DE 253 JPI BYTLM,4
02EA 254 JPI CPUTIM,4
02F6 255 JPI DFPFC,4
0302 256 JPI DFWSCNT,4
030E 257 JPI DIOCNT,4
031A 258 JPI DIOLM,4
0326 259 JPI DIRIO,4
0332 260 JPI EFCS,4
033E 261 JPI EFCU,4
034A 262 JPI EFWM,4
0356 263 JPI EXCVEC,4
0362 264 JPI FINALEXC,4
036E 265 JPI FILCNT,4
037A 266 JPI FILLM,4
0386 267 JPI FREPOVA,4
0392 268 JPI FREPIVA,4
039E 269 JPI GPGCNT,4
03AA 270 JPI IMAGNAME,128
03B6 271 JPI LOGINTIM,4
03C2 272 JPI OWNER,4
03CE 273 JPI PAGEFLTS,4
03DA 274 JPI PGFLQUOTA,4
03E6 275 JPI PID,4
03F2 276 JPI PPGCNT,4
03FE 277 JPI PRCCNT,4
040A 278 JPI PRCNAM,15
0416 279 JPI PROCPRIV,8
0422 280 JPI PRI,4
042E 281 JPI PRI8,4
043A 282 JPI STATE,4
0446 283 JPI STS,4
0452 284 JPI TMBU,4
```

SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.)^{K 8} 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 8
(1)

	045E	285	JPI TQCNT,4	
	046A	286	JPI VOLUMES,4	
	0476	287	JPI VIRTPEAK,4	
	0482	288	JPI WSAUTH,4	
	048E	289	JPI WSQUOTA,4	
	049A	290	JPI WSPEAK,4	
	04A6	291	JPI WSSIZE,4	
00000000	04B2	292	.LONG 0	; list terminator

20 54 53 45 54 53 59 53	04B6 294 JPI_GOOD:	: expected GETJPI results
0008	04B6 295	: Item name buffer offset
00000000	04B6 296 .ASCII /SYSTEST /	: ACCOUNT 00
00000000 1070BF EF	04BE 297 .WORD 8	: ACCOUNTL 08
00000001	04C0 298 JPI_GOOD SHRT:	
0002	04C0 299 .LONG 0	: CPULIM 0A
00000000 00000000	04C4 300 .WORD 4	: CPULIML 0E
0008	04C6 301 .QUAD JPI_PRIV_MASK	: CURPRIV 10
00000001	04CE 302 .WORD 8	: CURPRIVL 18
0002	04D0 303 .LONG 1	: GRP 1A
00000000 00000000	04D4 304 .WORD 2	: GRPL 1E
0008	04D6 305 .QUAD 00	: IMAGEPRIV 20
00000007	04DE 306 .WORD 8	: IMAGEPRIVL 28
0002	04E0 307 .LONG 7	: MEM 2A
00000008	04E4 308 .WORD 2	: MEML 2E
0002	04E6 309 .LONG 8	: PRCLM 30
00000014	04EA 310 .WORD 2	: PRCLML 38
0002	04EC 311 .LONG ^X14	: TQLM 3A
00010007	04F0 312 .WORD 2	: TQLML 3E
0004	04F2 313 .LONG ^X10007	: UIC 40
20 20 20 20 20 54 53 45 54 53 59 53	04F6 314 .WORD 4	: UICL 44
000C	04F8 315 .ASCII /SYSTEST /	: USERNAME 46
	0504 316 .WORD ^XC	: USERNAMEL 52
4E 49 24 53 59 53 0000050E'010E0000'	0506 317 IN:	
54 55 50	0506 318 .ASCID /SYS\$INPUT/	
55 4F 24 53 59 53 0000051F'010E0000'	0514 319 OUT:	
54 55 50 54	0517 320 .ASCID /SYS\$OUTPUT/	
52 45 24 53 59 53 00000531'010E0000'	0525 321 ERR:	
52 4F 52	0529 322 .ASCID /SYS\$ERROR/	
54 55 53 54 41 53 00000542'010E0000'	0537 323 IMAGE_NAME:	
45 58 45 2E 31 30	053A 324 .ASCID /SATSUT01.EXE/	
54 55 53 54 41 53 00000556'010E0000'	0548 325 PROC_NAME:	
35 33	054E 326 .ASCID /SATSUT35/	
00000565	055C 327 .BLKB 7	
00010007	055E 328 PROC_UIC:	: process UIC
	0565 329 .LONG ^X10007	

```
0569 331 ;
0569 332 .SBTTL R/W PSECT
00000000 333 .PSECT RWDATA,RD,WRT,NOEXE, LONG
0000 334
0000 335 ;PID:
00000000 336 .LONG 0 ; PID for this process
00000000 337 CURRENT_TC: .LONG 0 ; ptr to current test case
00000000 338 .ALIGN LONG ; put it on a long word boundry
0008 339 REG_SAVE_AREA:
00000044 340 .BLKL 15 ; register save area
0044 341 MOD_MSG_CODE:
007480D9 342 .LONG UETPS_SATSMS ; test module message code for putmsg
0048 343 TMN_ADDR:
00000000' 344 .ADDRESS TEST_MOD_NAME
004C 345 TMD_ADDR:
00000019' 346 .ADDRESS TEST_MOD_BEGIN
0050 347 PRVPRT:
00 348 .BYTE 0 ; protection return byte for SETPRT
00000000 00000000 349 PRIVMASK: .QUAD 0 ; priv. mask
0059 350 CHM_CONT: .LONG 0 ; change mode continue address
005D 351 RETADR: .BLKL 2 ; returned address's from SETPRT
0065 352 STATUS: .LONG 0
0069 353 MODE: .LONG 0 ; current mode string pointer
006D 354 REG: .ASCII \register R\
74 73 69 67 65 72 00000075'010E0000' 355 361
52 20 72 65 007B 362 REGNUM:
007F 363 .LONG 0 ; register number
00000000 007F 364 MSGL: .LONG 80 ; buffer desc.
0083 365 .ADDRESS BUF
00000050 0083 366 CRE:
00000133' 0087 367 $CREPRC PID1,0,0,0,0,0,QUOTA_LIST,-
008B 368 0,0,0,0,0 ; CREPRC parameter list
008B 369 GET: .GETJPI EFN=1,PIDADR=PID1,PRCNAM=TEST_MOD_NAME_D,ITMLST=GET_LIST
00C3 370 GET1: $GETJPI ITMLST=GET_LIST
00E3 371
00E3 372 ITEM_LIST:
0103 373 .BLKL 12
00000133 0103 374 BUF: .BLKB 80
0133 375 ML: .LONG 0 ; desc. for BUF_CHECK routine
0183 376 .ADDRESS GETBUF+8
00000000 0183 377 GETBUF: .LONG 132
00000193' 0187 378 .ADDRESS +4
018B 379 .BLKB 132
00000084 018B 380 MESSAGEL:
00000193' 018F 381 .LONG 0 ; message desc.
00000217 0193 382
0217 383
00000000 0217 384
```

00000133'	021B	387	.ADDRESS BUF	
	021F	388	SERV_NAME:	
00000000	021F	389	.LONG	0 ; service name pointer
	0223	390	MSGVEC1:	; PUTMSG message vector
00000003	0223	391	.LONG	3
00741130	0227	392	.LONG	UETPS_TEXT
00000001	022B	393	.LONG	1
00000000	022F	394	.LONG	0
	0233	395	IOSTAT:	
00000000 00000000	0233	396	.QUAD	0 ; IO status block
	023B	397	PID1:	
00000000	023B	398	.LONG	0 ; PID storage location
	023F	399	MBCHAN:	
0000	023F	400	.WORD	0 ; MBX channel location
	0241	401	MBXUN:	
0000	0241	402	.WORD	0 ; MBX unit number
	0243	403	IOSTATUS:	
0000024B	0243	404	.BLKL	2 ; MBX read IO status block
	024B	405	MBUF:	
000002AF	024B	406	.BLKB	100 ; MBX read buffer
	02AF	407	TEST_PID:	
00000000	02AF	408	.LONG	0 ; GETJPI parameter
	02B3	409	PRIVS:	
00000000 00000000	02B3	410	.QUAD	0 ; privilege mask

```
00000000 412 .PSECT SATSSS35, RD, WRT, EXE, LONG
0000 413 .SBTTL SATSSS35
0000 414 :++
0000 415 : FUNCTIONAL DESCRIPTION:
0000 416 :
0000 417 : After performing some initial housekeeping, such as
0000 418 : printing the module begin message and acquiring needed privileges,
0000 419 : the system services are tested in each of their normal conditions.
0000 420 : Detected failures are identified and an error message is printed
0000 421 : on the terminal. Upon completion of the test a success or fail
0000 422 : message is printed on the terminal.
0000 423 :
0000 424 : CALLING SEQUENCE:
0000 425 :
0000 426 : $ RUN SATSSS35 ... (DCL COMMAND)
0000 427 :
0000 428 : INPUT PARAMETERS:
0000 429 :
0000 430 : none
0000 431 :
0000 432 : IMPLICIT INPUTS:
0000 433 :
0000 434 : none
0000 435 :
0000 436 : OUTPUT PARAMETERS:
0000 437 :
0000 438 : none
0000 439 :
0000 440 : IMPLICIT OUTPUTS:
0000 441 :
0000 442 : Messages to SYS$OUTPUT are the only output from SATSSS35.
0000 443 : They are of the form:
0000 444 :
0000 445 : %UETP-S-SATSMS, TEST MODULE SATSSS35 BEGUN ... (BEGIN MSG)
0000 446 : %UETP-S-SATSMS, TEST MODULE SATSSS35 SUCCESSFUL ... (END MSG)
0000 447 : %UETP-E-SATSMS, TEST MODULE SATSSS35 FAILED ... (END MSG)
0000 448 : %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000 449 :
0000 450 : COMPLETION CODES:
0000 451 :
0000 452 : The SATSSS35 routine terminates with a $EXIT to the
0000 453 : operating system with a status code defined by UETPS_SATSMS.
0000 454 :
0000 455 : SIDE EFFECTS:
0000 456 :
0000 457 : none
0000 458 :
0000 459 : --
0000 460 :
0000 461 : TEST_START SATSSS35 ; let the test begin
```

```
0000 0000 .ENTRY SATSSS35,0
0004'CF 00 D4 0002 CLRL W^CURRENT_TC
00 DD 0006 PUSHL #0
0000'CF 00 DF 0008 PUSHAL W^TPID
00000000'GF 02 FB 000C CALLS #2,G^SYSSWAKE
00000000'GF 00 FB 0013 CALLS #0,G^SYSSHIBER
00009'CF 01 7F 001A PUSHAQ W^TEST_MOD_NAME_D
00000000'GF 01 FB 001E CALLS #1,G^SYSSSETPRN
004C'CF 001F'CF 30 0025 BSBW W^MOD_MSG_PRINT
0044'CF 03 00 01 DE 0028 MOVAL W^TEST_MOD_SUCC,W^TMD_ADDR
00 00 FO 002F INSV #SUCCESS,#0,#3,W^MOD_MSG_CODE
0BF3'CF 01 FB 0036 PUSHL #0
003D 0038 CALLS #1,W^REG_SAVE
003D 462 STP0:
003D 463 .SBTTL CREPRC TESTS
003D 464 :+
003D 465 $CREPRC tests
003D 466
003D 467 test the minimum quota all defaults subprocess with _S
003D 468
003D 469 :-
0069'CF 017C'CF DE 003D 470 MOVAL W^UM,W^MODE ; set the mode
021F'CF 016E'CF DE 0044 471 MOVAL W^CREPRC,W^SERV_NAME ; set the service name
0048 472 $CREMBX,S CHAN=W^MBCHAN,-
0048 473 LOGNAM=W^MBNAM ; make something to listen with
0062 474 $GETCHN,S CHAN=W^MBCHAN,-
0062 475 PRIBUF=W^GETBUF ; get the unit number
0241'CF 019F'CF B0 0078 476 MOVW W^GETBUF+8+DIBSW UNIT,W^MBXUN ; and save it
007F 477 $CREPRC,S QUOTA=W^QUOTA_LIST,-
007F 478 MBXUNT=W^MBXUN ; create a subprocess with _S
00A5 479 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 00A5
0BFD'CF 01 FB 00AB
56 00000000'8F D0 00B0 480 MOVL #RMS$_FNF,R6 ; set exit status code
00 57 D4 00B7 481 CLRL R7 ; disable PID checking this time
0D30'CF 00 FB 00B9 482 CALLS #0,W^CRE_CHECK ; check the process exit code
00BE 483 :+
00BE 484
00BE 485 test the PIDADR parameter with _G
00BE 486
00BE 487 :-
00BE 488 NEXT_TEST
00BE
0004'CF 01 D0 00BE STP1:
00 DD 00C3
0BF3'CF 01 FB 00C5
00B7'CF 0241'CF B0 00CA 489 MOVW W^MBXUN,W^CRE+CREPRCS_MBXUNT ; set the MBX unit number
00D1 490 $CREPRC,G W^CRE ; try _G and PIDADR param.
00DA 491 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 00DA
0BFD'CF 01 FB 00E0
00 57 D6 00E5 492 INCL R7 ; enable PID checking
0D30'CF 00 FB 00E7 493 CALLS #0,W^CRE_CHECK ; check the process exit code
56 00000000'8F D0 00EC 494 MOVL #SS$_NORMAL,R6 ; set expected status return
```

```

00F3 495 :+
00F3 496 :
00F3 497 : test the IMAGE param. with _S
00F3 498 :
00F3 499 :-
00F3 500 : NEXT_TEST
00F3
00F3 STP2:
0004'CF 02 DO 00F3 MOVL #2,W^CURRENT_TC
00F3 DD 00F8 PUSHL #0
0BF3'CF 01 FB 00FA CALLS #1,W^REG_SAVE
00FF 501 $CREPRC_S QUOTA=W^QUOTA_LIST,-
00FF 502 IMAGE=W^IMAGE_NAME,-
00FF 503 MBXUNT=W^MBXUNT,-
00FF 504 PIDADR=W^PID1 ; try _S with IMAGE param.
0129 505 FAIL_CHECK $$$_NORMAL ; check success
00000000'8F DD 0129 PUSHL $$$_NORMAL
0BFD'CF 01 FB 012F CALLS #1,W^REG_CHECK
0134 506 $WAKE_S PIDADR = W^PID1 ; cause process termination
0D30'CF 00 FB 0141 507 CALLS #0,W^CRE_CHECK ; check the process exit code
0146 508 :+
0146 509 :
0146 510 : test the INPUT param. with _G
0146 511 :
0146 512 :-
0146 513 : NEXT_TEST
0146
0146 STP3:
0004'CF 03 DO 0146 MOVL #3,W^CURRENT_TC
00F3'CF 00 DD 014B PUSHL #0
0BFD'CF 01 FB 014D CALLS #1,W^REG_SAVE
0093'CF 053A'CF DE 0152 514 MOVAL W^IMAGE_NAME,W^CRE+CREPRCS_IMAGE ; set image name
008F'CF 023B'CF DE 0159 515 MOVAL W^PID1,W^CRE+CREPRCS_PIDADR ; set the PID save address
0097'CF 0506'CF DE 0160 516 MOVAL W^IN,W^CRE+CREPRCS_INPUT ; set the INPUT param.
0167 517 $CREPRC_G W^CRE ; try it
0170 518 FAIL_CHECK $$$_NORMAL ; check success
00000000'8F DD 0170 PUSHL $$$_NORMAL
0BFD'CF 01 FB 0176 CALLS #1,W^REG_CHECK
017B 519 $WAKE_S PIDADR = W^PID1 ; cause process termination
0D30'CF 00 FB 0188 520 CALLS #0,W^CRE_CHECK ; check the process exit code
018D 521 :+
018D 522 :
018D 523 : test the OUTPUT param. with _S
018D 524 :
018D 525 :-
018D 526 : NEXT_TEST
018D
018D STP4:
0004'CF 04 DO 018D MOVL #4,W^CURRENT_TC
00F3'CF 00 DD 0192 PUSHL #0
0BFD'CF 01 FB 0194 CALLS #1,W^REG_SAVE
0199 527 $CREPRC_S PIDADR=W^PID1,-
0199 528 IMAGE=W^IMAGE_NAME,-
0199 529 INPUT=W^IN,-
0199 530 OUTPUT=W^OUT,-
0199 531 MBXUNT=W^MBXUNT,-
0199 532 QUOTA=W^QUOTA_LIST ; try _S with OUT param.

```

```
00000000'8F DD 01C7 533 FAIL_CHECK SSS_NORMAL ; chec success
0BFD'CF 01 FB 01C7 PUSHL #SS$ NORMAL
0D30'CF 00 FB 01CD CALLS #1,W^REG_CHECK
01D2 534 $WAKE_S PIDADR = W^PID1 ; cause process termination
01DF 535 CALLS #0,W^CRE_CHECK ; check process exit code
01E4 536 :+
01E4 537 : test ERROR param. with _G
01E4 538 :
01E4 539 :
01E4 540 :-
01E4 541 NEXT_TEST
01E4
01E4 STP5:
0004'CF 05 DO 01E4 MOVL #5,W^CURRENT_TC
0000'CF 00 DD 01E9 PUSHL #0
0BF3'CF 01 FB 01EB CALLS #1,W^REG_SAVE
009B'CF 0517'CF DE 01F0 542 MOVAL W^OUT,W^CRE+CREPRCS_OUTPUT ; set the output param.
009F'CF 0529'CF DE 01F7 543 MOVAL W^ERR,W^CRE+CREPRCS_ERROR ; set the error output param
01FE 544 $CREPRC G W^CRE ; try _G with ERROR param
0207 545 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 0207 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 020D CALLS #1,W^REG_CHECK
0D30'CF 00 FB 0212 546 $WAKE_S PIDADR = W^PID1 ; cause process termination
021F 547 CALLS #0,W^CRE_CHECK ; check process exit code
0224 548 :+
0224 549 : test PRVADR param with _S
0224 550 :
0224 551 :
0224 552 :-
0224 553 NEXT_TEST
0224
0224 STP6:
0004'CF 06 DO 0224 MOVL #6,W^CURRENT_TC
0000'CF 00 DD 0229 PUSHL #0
0BF3'CF 01 FB 022B CALLS #1,W^REG_SAVE
0230 554 $CREPRC_S PIDADR=W^PID1,-
0230 555 IMAGE=W^IMAGE_NAME,-
0230 556 INPUT=W^IN,-
0230 557 OUTPUT=W^OUT,-
0230 558 ERROR=W^ERR,-
0230 559 PRVADR=W^PRIVS,-
0230 560 MBXUNT=W^MBXUN,-
0230 561 QUOTA=W^QUOTA_LIST ; try _S with PRVADR param
0262 562 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 0262 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 0268 CALLS #1,W^REG_CHECK
0D30'CF 00 FB 026D 563 $WAKE_S PIDADR = W^PID1 ; cause process termination
027A 564 CALLS #0,W^CRE_CHECK ; check image exit status
027F 565 :+
027F 566 : test PRCNAM param with _G
027F 567 :
027F 568 :
027F 569 :-
027F 570 NEXT_TEST
027F
0004'CF 07 DO 027F STP7:
MOVL #7,W^CURRENT_TC
```

00	DD	0284		PUSHL	#0	
009F'CF	01	FB	0286	CALLS	#1,W*REG_SAVE	
00A3'CF	0529'CF	DE	0288	MOVAL	W*ERR,W*CRE+CREPRC\$_ERROR	: set the ERROR param.
00AB'CF	02B3'CF	DE	0292	MOVAL	W*PRIVS,W*CRE+CREPRC\$ _PRVADR	: set the PRVADR param.
	054E'CF	DE	0299	MOVAL	W*PROC_NAME,W*CRE+CREPRC\$ _PRCNAM	: set the process name
			02A0	\$CREPRC	G W*CRE	: try G with a PRCNAM
			02A9	FAIL_CHECK	SS\$ _NORMAL	: check success
00000000'8F	DD	02A9		PUSHL	#SS\$ _NORMAL	
0BFD'CF	01	FB	02AF	CALLS	#1,W*REG_CHECK	
021F'CF	0175'CF	DE	02B4	MOVAL	W*GETJPI,W*SERV_NAME	: set service name
			02BB	\$GETJPI_S	PIDADR = W*PID1,-	
			02BB	ITMLST	= W*GET_LIST	: get the process name
			02D2	FAIL_CHECK	SS\$ _NORMAL	: check success
00000000'8F	DD	02D2		PUSHL	#SS\$ _NORMAL	
0BFD'CF	01	FB	02D8	CALLS	#1,W*REG_CHECK	
021F'CF	016E'CF	DE	02DD	MOVAL	W*CREPRC,W*SERV_NAME	: set service name
0556'CF	01A2'CF	0F	29	CMPC3	#15,W*PRCNAM,W*PROC_NAME+8	: correct process name?
	0109'CF	09	13	BEQL	10\$: br if OK
	0C3F'CF	01	DF	PUSHL	W*PNS	: push string variable
			FB	CALLS	#1,W*PRINT_FAIL	: print the failure
			02F7			
			02F7	10\$:		
0D30'CF	00	FB	0304	\$WAKE_S	PIDADR = W*PID1	: cause process termination
			0309	CALLS	#0,W*CRE_CHECK	: check image exit status
			0309	++		
			0309	589		
			0309	590	test BASPRI with _S and a lower priority	
			0309	591		
			0309	592		
			0309	593		
			0309	:-		
			0309		NEXT_TEST	
0004'CF	08	DO	0309	STP8:		
	00	DD	030E	MOVL	#8,W*CURRENT_TC	
0BFD'CF	01	FB	0310	PUSHL	#0	
			0315	CALLS	#1,W*REG_SAVE	
			0315	\$CREPRC_S	PIDADR = W*PID1,-	
			0315		IMAGE = W*IMAGE_NAME,-	
			0315		INPUT = W*IN,-	
			0315		OUTPUT = W*OUT,-	
			0315		ERROR = W*ERR,-	
			0315		BASPRI = #1,-	
			0315		PRVADR = W*PRIVS,-	
			0315		MBXUNT = W*MBXUN,-	
			0315		QUOTA = W*QUOTA_LIST	: try all that
			0347	FAIL_CHECK	SS\$ _NORMAL	: check success
00000000'8F	DD	0347		PUSHL	#SS\$ _NORMAL	
0BFD'CF	01	FB	034D	CALLS	#1,W*REG_CHECK	
021F'CF	0175'CF	DE	0352	MOVAL	W*GETJPI,W*SERV_NAME	: set service name
			0359	\$GETJPI_S	PIDADR = W*PID1,-	
			0359	ITMLST	= W*GET_LIST	: get the base priority
			0370	FAIL_CHECK	SS\$ _NORMAL	: check success
00000000'8F	DD	0370		PUSHL	#SS\$ _NORMAL	
0BFD'CF	01	FB	0376	CALLS	#1,W*REG_CHECK	
021F'CF	016E'CF	DE	037B	MOVAL	W*CREPRC,W*SERV_NAME	: set service name
	01C3'CF	01	D1	CMPL	#1,W*PRIB	: is it correct?
		0F	13	BEQL	20\$: br if OK
	01C3'CF	DD	0387	PUSHL	W*PRIB	: push received
		DD	0389	PUSHL	#1	: push expected
			038D			

	00F8'CF	DF	038F	613	PUSHAL W^BP	; push str variable
	0C3F'CF	FB	0393	614	CALLS #3,W^PRINT_FAIL	; print the failure
			0398	615	20\$:	
	0D30'CF	FB	03A5	616	\$WAKE_S PIDADR = W^PID1	; cause process termination
			03AA	617	CALLS #0,W^CRE_CHECK	; check image exit status
			03AA	618	::+	
			03AA	619	:	
			03AA	620	: test BASPRI with _S and a higher priority	
			03AA	621	:	
			03AA	622	:-	
			03AA	623	NEXT_TEST	
			03AA			
	0004'CF	DD	03AA		STP9:	
		DD	03AF		MOVL #9,W^CURRENT_TC	
	0BF3'CF	FB	03B1		PUSHL #0	
			03B6	624	CALLS #1,W^REG_SAVE	
59	00000000'9F	DD	03D3	625	MODE TO,25\$,KRNL,NOREGS	; kernal mode to access PHD
	0051'CF	DE	03DA	626	MOVL @#CTL\$GL_PHD,R9	; get process header address
			03DF	627	MOVAL PHD\$Q_PRIVMSK(R9),W^PRIVMASK	; get priv mask address
			03E0	628	MODE FROM,25\$; get back to user mode
		DD	0400	629	PRIV ADD,SETPRI	; add SETPRI priv
	0BF3'CF	FB	0402	630	PUSHL #0	; push a dummy parameter
			0407	631	CALLS #1,W^REG_SAVE	; save the registers
			0407	632	\$CREPRC_S PIDADR = W^PID1,-	
			0407	633	IMAGE = W^IMAGÉ_NAME,-	
			0407	634	INPUT = W^IN,-	
			0407	635	OUTPUT = W^OUT,-	
			0407	636	ERROR = W^ERR,-	
			0407	637	BASPRI = #4,-	
			0407	638	PRVADR = W^PRIVS,-	
			0407	639	MBXUNT = W^MBXUN,-	
			0439	640	QUOTA = W^QUOTA_LIST	; try _S higher priority
	00000000'8F	DD	0439		FAIL_CHECK \$\$\$_NORMAL	; check success
	0BFD'CF	FB	043F		PUSHL #\$\$\$ NORMAL	
021F'CF	0175'CF	DE	0444	641	CALLS #1,W^REG CHECK	
			044B	642	MOVAL W^GETJPI,W^SERV_NAME	; set the service name
			044B	643	\$GETJPI_S PIDADR = W^PID1,-	
			0462	644	ITMLST = W^GET_LIST	; get the base priority
	00000000'8F	DD	0462		FAIL_CHECK \$\$\$_NORMAL	; check success
	0BFD'CF	FB	0468		PUSHL #\$\$\$ NORMAL	
021F'CF	016E'CF	DE	046D	645	CALLS #1,W^REG CHECK	
	01C3'CF	D1	0474	646	MOVAL W^CREPRC,W^SERV_NAME	; reset the service name
		13	0479	647	CNPL #4,W^PRIB	; is the priority OK?
	01C3'CF	DD	047B	648	BEQL 30\$; br if OK
		DD	047F	649	PUSHL W^PRIB	; push received
	00F8'CF	DF	0481	650	PUSHL #4	; push expected
	0C3F'CF	FB	0485	651	PUSHAL W^BP	; push the str variable
			048A	652	CALLS #3,W^PRINT_FAIL	; print the failure
			048A	653	30\$:	
	0D30'CF	FB	0497	654	\$WAKE_S PIDADR = W^PID1	; cause process termination
			049C	655	CALLS #0,W^CRE_CHECK	; check image exit status
			049C	656	::+	
			049C	657	:	
			049C	658	: test detached process	
			049C	659	:	
			049C	660	:-	
					NEXT_TEST	

```
0004'CF 0A DO 049C
0000'CF 00 DD 049C
0BF3'CF 01 FB 04A1
04A8 661
04A8 662
04A8 663
04A8 664
04A8 665
04A8 666
04A8 667
04A8 668
04A8 669
04A8 670
04DC 671
00000000'8F DD 04DC
0BFD'CF 01 FB 04E2
021F'CF 0175'CF DE 04E7 672
04EE 673
04EE 674
0505 675
00000000'8F DD 0505
0BFD'CF 01 FB 050B
021F'CF 016E'CF DE 0510 676
0565'CF 003C'CF D1 0517 677
11 13 051E 678
003C'CF DD 0520 679
0565'CF DD 0524 680
0144'CF DF 0528 681
0C3F'CF 03 FB 052C 682
0531 683
0531 684
053E 685
0543 686
0543 687
0543 688
0543 689
0543 690
0543 691
0004'CF 0B DO 0543
0000'CF 00 DD 0543
0BF3'CF 01 FB 0548
054A 692
054F 693
59 00000000'9F DO 056C 694
0051'CF 69 DE 0573 695
0578 696
0579 697
0599 698
0589 699
00 DD 05D9 700
0BFD'CF 01 FB 05DB 701
021F'CF 0175'CF DE 05E0 702
05E7 703
05FC 703
```

STP10:

```
MOVL #10,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
SCREPRC_S PIDADR = W^PIDT,-
IMAGE = W^IMAGE_NAME,-
INPUT = W^IN,-
OUTPUT = W^OUT,-
ERROR = W^ERR,-
BASPRI = #2,-
PRVADR = W^PRIVS,-
MBXUNT = W^MBXUN,-
QUOTA = W^QUOTA_LIST,-
UIC = W^PROC_UIC
FAIL_CHECK SSS_NORMAL
PUSHL #SSS_NORMAL
CALLS #1,W^REG_CHECK
MOVAL W^GETJPI,W^SERV_NAME
$GETJPI_S PIDADR = W^PID1,-
ITMLST = W^GET_LIST
FAIL_CHECK SSS_NORMAL
PUSHL #SSS_NORMAL
CALLS #1,W^REG_CHECK
MOVAL W^CREPRC,W^SERV_NAME
CMPL W^UIC,W^PROC_UIC
BEQL 40$
PUSHL W^UIC
PUSHL W^PROC_UIC
PUSHAL W^UIC MSG
CALLS #3,W^PRINT_FAIL
$WAKE_S PIDADR = W^PID1
CALLS #0,W^CRE_CHECK
40$:
:+
:-
test the STSFLG's _S with all set
NEXT_TEST
```

STP11:

```
MOVL #11,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
MODE TO,45$,KRNL,NOREGS
MOVL @#CTL$GL PHD,R9
MOVAL PHD$Q PRIVMSK(R9),W^PRIVMASK
MODE FROM,45$
PRIV ADD,PSWAPM
PRIV ADD,NOACNT
PRIV ADD,NETMBX
PUSHL #0
CALLS #1,W^REG_SAVE
MOVAL W^GETJPI,W^SERV_NAME
$GETJPI_S ITMLST = W^GET_LIST
FAIL_CHECK SSS_NORMAL
```

: try _S and all this
: check success
: set service name
: get the process UIC
: check success
: reset the service name
: is the UIC correct?
: br if OK
: push received
: push expected
: push the string variable
: print the failure
: cause process termination
: check the process exit status
: kernel mode to access PHD
: get process header address
: get priv mask address
: get back to user mode
: add PSWAPM priv
: add NOACNT priv
: add NETMBX priv
: push a dummy param
: save a reg snap shot
: set service name
: get the current process privs
: check success

```
00000000'8F DD 05FC
0BFD'CF 01 FB 0602
021F'CF 016E'CF DE 0607 704
02B3'CF 0010'CF 7D 060E 705
0615 706
0615 707
0615 708
0615 709
0615 710
0615 711
0615 712
0615 713
0615 714
0615 715
0615 716
064D 717
00000000'8F DD 064D
0BFD'CF 01 FB 0653
021F'CF 0175'CF DE 0658 718
065F 719
065F 720
0676 721
00000000'8F DD 0676
0BFD'CF 01 FB 067C
021F'CF 016E'CF DE 0681 722
000001CF'EF FFC739FF 8F CA 0688 723
0693 724
0038C600 8F 01CF'CF D1 0693 725
13 13 069C 726
01CF'CF DD 069E 727
0038C600 8F DD 06A2 728
0134'CF DF 06A8 729
0C3F'CF 03 FB 06AC 730
06B1 731 50$:
06B1 732
06BE 733
0D30'CF 00 D4 06C0 734
56 00000000'8F D0 06C5 735
06CC 736 :+
06CC 737 :
06CC 738 : test the STSFLG's _G all clear
06CC 739 :
06CC 740 :-
06CC 741
NEXT_TEST
STP12:
0004'CF 0C D0 06CC
00 DD 06D1
0BF3'CF 01 FB 06D3
00A3'CF 02B3'CF DE 06D8 742
00AB'CF 054E'CF DE 06DF 743
00AF'CF 02 D0 06E6 744
00B3'CF 0565'CF D0 06EB 745
06F2 746
021F'CF 0175'CF DE 06FB 747
0702 748
0702 749

PUSHL #SS$ NORMAL
CALLS #1,W*REG CHECK
MOVAL W*CREPRC,W*SERV_NAME ; set the service name
MOVQ W*CURPRIV,W*PRIVS ; set the detached process privs
SCREPRC_S PIDADR = W*PID1,-
IMAGE = W*IMAGE_NAME,-
INPUT = W*IN,-
OUTPUT = W*OUT,-
ERROR = W*ERR,-
BASPRI = #2,-
PRVADR = W*PRIVS,-
MBXUNT = W*MBXUN,-
QUOTA = W*QUOTA_LIST,-
UIC = W*PROC_OIC,-
STSFLG = #*XFF ; try every thing _S
; check success
FAIL_CHECK SS$ NORMAL
PUSHL #SS$ NORMAL
CALLS #1,W*REG CHECK
MOVAL W*GETJPI,W*SERV_NAME ; set service name
$GETJPI_S PIDADR = W*PID1,-
ITMLST = W*GET_LIST ; get process status flags
; check success
FAIL_CHECK SS$ NORMAL
PUSHL #SS$ NORMAL
CALLS #1,W*REG CHECK
MOVAL W*CREPRC,W*SERV_NAME ; reset service name
BICL #JPI_STS_NMASK,STS ; clear out any extraneous
; bits set by the Swapper
; flags OK?
; br if OK
; push received
; push expected
; push str variable
; print the failure
CMPL W*STS,#JPI_STS_MASK
BEQL 50$
PUSHL W*STS
PUSHL #JPI_STS_MASK
PUSHAL W*STSFLGS
CALLS #3,W*PRINT_FAIL
$DELPRI_S PIDADR = W*PID1
CLRL R6
CALLS #0,W*CRE CHECK ; needed for bit 5 in STS
; set expected status return
; check image exit status
; reset the expected status return
MOVL #SS$ NORMAL,R6
;+
; test the STSFLG's _G all clear
;-
NEXT_TEST
STP12:
MOVL #12,W*CURRENT_TC
PUSHL #0
CALLS #1,W*REG SAVE
MOVAL W*PRIVS,W*CRE+CREPRCS PRVADR ; setup PRVADR parameter
MOVAL W*PROC NAME,W*CRE+CREPRCS_PRCNAM ; setup PRCNAM parameter
MOVL #2,W*CRE+CREPRCS BASPRI ; setup BASPRI parameter
MOVL W*PROC UIC,W*CRE+CREPRCS_UIC ; setup UIC parameter
SCREPRC_G W*CRE ; try it all _G
MOVAL W*GETJPI,W*SERV_NAME ; set service name
$GETJPI_S PIDADR = W*PID1,-
ITMLST = W*GET_LIST ; get the process status flags
```

```

00000000'8F DD 0719 750
0BFD'CF 01 FB 071F
021F'CF 016E'CF DE 0724 751
000001CF'EF FFC739FF 8F CA 072B 752
                                0736 753
00 01CF'CF D1 0736 754
                                0F 13 073B 755
                                01CF'CF DD 073D 756
                                00 DD 0741 757
                                0134'CF DF 0743 758
0C3F'CF 03 FB 0747 759
                                074C 760 60$:
0D30'CF 00 FB 074C 761
                                0759 762

```

FAIL_CHECK SS\$_NORMAL

```

PUSHL #SS$ NORMAL
CALLS #1,W^REG CHECK
MOVAL W^CREPRC,W^SERV_NAME
BICL #JPI_STS_NMASK,STS

```

```

CMPL W^STS,#^X0
BEQL 60$
PUSHL W^STS
PUSHL #^X0
PUSHAL W^STSFLGS
CALLS #3,W^PRINT_FAIL

```

```

$WAKE_S PIDADR = W^PID1
CALLS #0,W^CRE_CHECK

```

; check success

```

; set the service name
; clear out any extraneous
; bits set by the Swapper
; are they all OK?
; br if OK
; push received
; push expected
; push str variable
; print the failure

```

```

; cause process termination
; check image exit status

```

```
075E 764 .SBTTL GETJPI TESTS
075E 765 :+
075E 766 :
075E 767 $GETJPI tests
075E 768 :
075E 769 test the default case with all items _S
075E 770 :-
075E 771 :
075E 772 NEXT_TEST
075E
075E STP13:
075E      MOVL    #13,W^CURRENT_TC
0763      PUSHL   #0
0765      CALLS   #1,W^REG_SAVE
076A 773      MOVAL  W^GETJPI,W^SERV_NAME      : set service name
0771 774      MOVAL  W^UM,W^MODE                : set the mode
0778 775      PRIV   ADD,SETPRV                  : get ready to set privileges
0798 776      $SETPRV_S ENBFLG = #1,-           : to known set for testing
0798 777      PRVADR = PRVMASK                   : enable expected privileges
07AB 778      $SETPRV_S PRVADR = NPRVMASK       : disable any extra privileges
07BE 779      FAIL_CHECK SSS_NORMAL             : set in SYSTEST account
07BE 780      PUSHL   #SS$ NORMAL                : check success
07BE 781      CALLS   #1,W^REG_CHECK
07C4 782      $GETJPI S ITMLST = W^GET_LIST      : try S
07C9 783      FAIL_CHECK SSS_NORMAL             : check success
07DE 784      PUSHL   #SS$ NORMAL
07E4 785      CALLS   #1,W^REG_CHECK
07E9 786      MOVAL  W^ACCOUNT,R6                : set questionable data adr
07EE 787      MOVAL  W^JPI_GOOD,R7              : set good data adr
58 07F3 788      MOVL   #JPI_CIST_SIZE,R8       : set the byte count
07FA 789      CALLS   #0,W^JPI_CHECK           : check the results
07FF 790      :+
07FF 791      test _G default case with all items
07FF 792      :-
07FF 793      NEXT_TEST
07FF
07FF STP14:
07FF      MOVL    #14,W^CURRENT_TC
0804      PUSHL   #0
0806      CALLS   #1,W^REG_SAVE
080B 794      $GETJPI G W^GET1                  : try G
0814 795      FAIL_CHECK SSS_NORMAL             : check success
0814      PUSHL   #SS$ NORMAL
081A      CALLS   #1,W^REG_CHECK
081F 796      CALLS   #0,W^JPI_CHECK           : check the results
0824 797      :+
0824 798      test local EFN
0824 799      :-
0824 800      NEXT_TEST
0824 801      :-
0824 802      NEXT_TEST
0824
0824 STP15:
```

```
0004'CF 0F DO 0824          MOVL #15,W^CURRENT_TC
                                PUSHL #0
                                CALLS #1,W^REG_SAVE
0BF3'CF 01 FB 0829          MOVAL W^CREPRC,W^SERV_NAME          ; set service name
021F'CF 016E'CF DE 082B          $CREPRC_S QUOTA = W^QUOTA_LIST,-
                                IMAGE = W^IMAGE_NAME,-
                                PIDADR = W^PID1,-
                                PRCNAM = W^PROC_NAME
                                FAIL_CHECK SSS_NORMAL          ; create the target process
                                ; check for success
00000000'8F DD 0861          PUSHL #SS$ NORMAL
0BFD'CF 01 FB 0867          CALLS #1,W^REG_CHECK
021F'CF 0175'CF DE 086C          MOVAL W^GETJPI,W^SERV_NAME          ; reset the service name
                                $GETJPI_S EFN = #1,-
                                ITMLST = W^GET_LIST
                                FAIL_CHECK SSS_NORMAL          ; try S with EFN
                                ; check success
00000000'8F DD 0888          PUSHL #SS$ NORMAL
0BFD'CF 01 FB 088E          CALLS #1,W^REG_CHECK
0DDA'CF 00 FB 0893          $WAITFR_S EFN = #1          ; wait for completion
00C7'CF 01 DO 089C          CALLS #0,W^JPI_CHECK          ; check the results
00CB'CF 023B'CF DE 08A1          MOVL #1,W^GET+GETJPI$ EFN          ; set the EFN
                                MOVAL W^PID1,W^GET+GETJPI$_PIDADR
                                $GETJPI G W^GET          ; set the target process PID
                                ; try G with target process
                                FAIL_CHECK SSS_NORMAL          ; check success
                                PUSHL #SS$ NORMAL
                                CALLS #1,W^REG_CHECK
                                $WAITFR_S EFN = #1          ; wait for completion
                                CALLS #0,W^JPI_CHECK          ; check the results
                                ;+
                                ; test common EFN with _S
                                ;-
                                NEXT_TEST
                                STP16:
0004'CF 10 DO 08CF          MOVL #16,W^CURRENT_TC
                                DD 08D4          PUSHL #0
                                0BF3'CF 01 FB 08D6          CALLS #1,W^REG_SAVE
                                08DB 827          $ASCEFC_S EFN = #65,-
                                08DB 828          NAME = W^EFC_NAME          ; get a common EF
                                08F0 829          $GETJPI_S EFN = #65,-
                                08F0 830          ITMLST = W^GET_LIST          ; try S with CEFC
                                0909 831          FAIL_CHECK SSS_NORMAL          ; check success
                                DD 0909          PUSHL #SS$ NORMAL
                                0BFD'CF 01 FB 090F          CALLS #1,W^REG_CHECK
                                0914 832          $WAITFR_S EFN = #65          ; wait for completion
                                0DDA'CF 00 FB 0921          CALLS #0,W^JPI_CHECK          ; check results
                                00C7'CF 00000041 8F DO 0926          MOVL #65,W^GET+GETJPI$_EFN          ; set the common EFC
                                092F 834          $GETJPI G W^GET          ; try G, CEFC, and target process
                                0938 835          FAIL_CHECK SSS_NORMAL          ; check for success
                                DD 0938          PUSHL #SS$ NORMAL
                                0BFD'CF 01 FB 093E          CALLS #1,W^REG_CHECK
                                0943 837          $WAITFR_S EFN = #65          ; wait for completion
                                0DDA'CF 00 FB 0950          CALLS #0,W^JPI_CHECK          ; check the results
                                0955 838          $DACEFC_S EFN = #65          ; release the CEFC
                                0962 839
                                0962 840
                                0962 841 ;+
```

```
0962 842 :  
0962 843 : test PIDADR  
0962 844 :  
0962 845 :-  
0962 846 NEXT_TEST  
0962  
0962 STP17:  
0962      MOVL    #17,W^CURRENT_TC  
0967      PUSHL  #0  
0969      CALLS  #1,W^REG_SAVE  
096E 847 $GETJPI_S EFN = #2,-  
096E 848      PIDADR = W^PID1,-  
096E 849      ITMLST = W^GET_LIST      : try S with PID  
0985 850 FAIL_CHECK SSS_NORMAL      : check success  
0985      PUSHL  #SSS_NORMAL  
098B      CALLS  #1,W^REG_CHECK  
0990 851 $WAITFR_S EFN = #2      : wait for completion  
0999 852 CALLS  #0,W^JPI_CHECK      : check the results  
099E 853 :+  
099E 854 : test PRCNAM  
099E 855 :  
099E 856 :-  
099E 857  
099E 858 NEXT_TEST  
099E  
099E STP18:  
099E      MOVL    #18,W^CURRENT_TC  
09A3      PUSHL  #0  
09A5      CALLS  #1,W^REG_SAVE  
09AA 859 $GETJPI_S EFN = #3,-  
09AA 860      PRCNAM = W^PROC_NAME,-  
09AA 861      ITMLST = W^GET_LIST      : try S with PRCNAM  
09C1 862 FAIL_CHECK SSS_NORMAL      : check success  
09C1      PUSHL  #SSS_NORMAL  
09C7      CALLS  #1,W^REG_CHECK  
09CC 863 $WAITFR_S EFN = #3      : wait for completion  
09D5 864 CALLS  #0,W^JPI_CHECK      : check the results  
09DA 865 $GETJPI_S EFN = #16,-  
09DA 866      PRCNAM = W^TEST_MOD_NAME_D,-  
09DA 867      ITMLST = W^GET_LIST      : try S with PRCNAM on self  
09F1 868 FAIL_CHECK SSS_NORMAL      : check success  
09F1      PUSHL  #SSS_NORMAL  
09F7      CALLS  #1,W^REG_CHECK  
09FC 869 $WAITFR_S EFN = #16      : wait for completion  
0A05 870 CALLS  #0,W^JPI_CHECK      : check the results  
0A0A 871 :+  
0A0A 872 : test IOSB  
0A0A 873 :  
0A0A 874 :-  
0A0A 875  
0A0A 876 NEXT_TEST  
0A0A  
0A0A STP19:  
0A0A      MOVL    #19,W^CURRENT_TC  
0A0F      PUSHL  #0  
0A11      CALLS  #1,W^REG_SAVE  
0A16 877 $GETJPI_S EFN = #4,-
```

```
00000000'8F DD 0A16 878 IOSB = W^IOSTAT,-
0BFD'CF 01 FB 0A16 879 PRCNAM = W^IMAGE_NAME,-
0DDA'CF 00 FB 0A16 880 PIDADR = W^PID1,-
0A31 881 ITMLST = W^GET_LIST ; try all this stuff
0A31 882 FAIL_CHECK SS$ NORMAL ; check success
0A31 883 PUSHL #SS$ NORMAL
0A37 884 CALLS #1,W^REG_CHECK
0A3C 885 $WAITFR_S EFN = #4 ; wait for completion
0A45 886 CALLS #0,W^JPI_CHECK ; check the results
0A4A 887 :+
0A4A 888 : test ASTADR and ASTPRM _S
0A4A 889 :-
0A4A 890 NEXT_TEST
0A4A 891 STP20:
0A4A 892 MOVL #20,W^CURRENT_TC
0A4F 893 PUSHL #0
0A51 894 CALLS #1,W^REG_SAVE
0A56 895 $SETAST_S ENBFLG = #0 ; disable AST's
0A5F 896 $GETJPI_S ASTADR = B^20$,-
0A5F 897 ASTPRM = #5,-
0A5F 898 IOSB = W^IOSTAT,-
0A5F 899 PRCNAM = W^IMAGE_NAME,-
0A5F 900 PIDADR = W^PID1,-
0A5F 901 ITMLST = W^GET_LIST ; try an AST
0A7D 898 FAIL_CHECK SS$ NORMAL ; check success
0A7D 899 PUSHL #SS$ NORMAL
0A83 900 CALLS #1,W^REG_CHECK
0A88 899 $SETAST_S ENBFLG = #1 ; let er rip
0A91 900 $HIBER_S ; wait here for completion
0A98 901 BRB 40$ ; jump over the AST routine
0A9A 902 20$:
0A9A 903 .WORD ^M<R2,R3,R4>
0A9C 904 CMPL #5,4(AP) ; is this the right param
0AA0 905 BEQL 30$ ; br if OK
0AA2 906 PUSHL 4(AP) ; push the received
0AA5 907 PUSHL #5 ; push expected
0AA7 908 PUSHAL W^AST_PARAM ; push the string variable
0AAB 909 CALLS #3,W^PRINT_FAIL ; print the failure
0AB0 910 30$:
0AB0 911 CALLS #0,W^JPI_CHECK ; check the results
0AB5 912 $WAKE_S ; time to wake up
0AC0 913 RET ; return
0AC1 914 40$:
0AC1 915 :+
0AC1 916 : test ASTADR and ASTPRM _G to test all offset definitions
0AC1 917 :-
0AC1 918
0AC1 919
0AC1 920 NEXT_TEST
0AC1 921 STP21:
0AC1 922 MOVL #21,W^CURRENT_TC
0AC6 923 PUSHL #0
0AC8 924 CALLS #1,W^REG_SAVE
```

```
00C7'CF 0C DO OACD 921
00CB'CF 023B'CF DE OAD6 922
00CF'CF 053A'CF DE OADB 923
00D7'CF 0233'CF DE OAE2 924
00DB'CF 25'AF DE OAE9 925
00DF'CF FFFFFFFF 8F DO OAF0 926
OAF6 927
OAF7 928
OAF8 929
OAF9 930
OADA 931
OADB 932
OACD 933
OACE 934
OACF 935
OAC0 936
OAC1 937
OAC2 938
OAC3 939
OAC4 940
OAC5 941
OAC6 942
OAC7 943
OAC8 944
OAC9 945
OACA 946
OACB 947
OACC 948
OACD 949
OACE 950
OACF 951
OAC0 952
OAC1 953
OAC2 954
OAC3 955
OAC4 956
OAC5 957
OAC6 958
OAC7 959
OAC8 960
OAC9 961
OACA 962
OACB 963
OACC 964
OACD 965
OACE 966
OACF 967
OAC0 968
OAC1 969
OAC2 970
OAC3 971
OAC4 972
OAC5 973
OAC6 974
OAC7 975
OAC8 976
OAC9 977
OACA 978
OACB 979
OACC 980
OACD 981
OACE 982
OACF 983
OAC0 984
OAC1 985
OAC2 986
OAC3 987
OAC4 988
OAC5 989
OAC6 990
OAC7 991
OAC8 992
OAC9 993
OACA 994
OACB 995
OACC 996
OACD 997
OACE 998
OACF 999
OAC0 1000
OAC1 1001
OAC2 1002
OAC3 1003
OAC4 1004
OAC5 1005
OAC6 1006
OAC7 1007
OAC8 1008
OAC9 1009
OACA 1010
OACB 1011
OACC 1012
OACD 1013
OACE 1014
OACF 1015
OAC0 1016
OAC1 1017
OAC2 1018
OAC3 1019
OAC4 1020
OAC5 1021
OAC6 1022
OAC7 1023
OAC8 1024
OAC9 1025
OACA 1026
OACB 1027
OACC 1028
OACD 1029
OACE 1030
OACF 1031
OAC0 1032
OAC1 1033
OAC2 1034
OAC3 1035
OAC4 1036
OAC5 1037
OAC6 1038
OAC7 1039
OAC8 1040
OAC9 1041
OACA 1042
OACB 1043
OACC 1044
OACD 1045
OACE 1046
OACF 1047
OAC0 1048
OAC1 1049
OAC2 1050
OAC3 1051
OAC4 1052
OAC5 1053
OAC6 1054
OAC7 1055
OAC8 1056
OAC9 1057
OACA 1058
OACB 1059
OACC 1060
OACD 1061
OACE 1062
OACF 1063
OAC0 1064
OAC1 1065
OAC2 1066
OAC3 1067
OAC4 1068
OAC5 1069
OAC6 1070
OAC7 1071
OAC8 1072
OAC9 1073
OACA 1074
OACB 1075
OACC 1076
OACD 1077
OACE 1078
OACF 1079
OAC0 1080
OAC1 1081
OAC2 1082
OAC3 1083
OAC4 1084
OAC5 1085
OAC6 1086
OAC7 1087
OAC8 1088
OAC9 1089
OACA 1090
OACB 1091
OACC 1092
OACD 1093
OACE 1094
OACF 1095
OAC0 1096
OAC1 1097
OAC2 1098
OAC3 1099
OAC4 1100
OAC5 1101
OAC6 1102
OAC7 1103
OAC8 1104
OAC9 1105
OACA 1106
OACB 1107
OACC 1108
OACD 1109
OACE 1110
OACF 1111
OAC0 1112
OAC1 1113
OAC2 1114
OAC3 1115
OAC4 1116
OAC5 1117
OAC6 1118
OAC7 1119
OAC8 1120
OAC9 1121
OACA 1122
OACB 1123
OACC 1124
OACD 1125
OACE 1126
OACF 1127
OAC0 1128
OAC1 1129
OAC2 1130
OAC3 1131
OAC4 1132
OAC5 1133
OAC6 1134
OAC7 1135
OAC8 1136
OAC9 1137
OACA 1138
OACB 1139
OACC 1140
OACD 1141
OACE 1142
OACF 1143
OAC0 1144
OAC1 1145
OAC2 1146
OAC3 1147
OAC4 1148
OAC5 1149
OAC6 1150
OAC7 1151
OAC8 1152
OAC9 1153
OACA 1154
OACB 1155
OACC 1156
OACD 1157
OACE 1158
OACF 1159
OAC0 1160
OAC1 1161
OAC2 1162
OAC3 1163
OAC4 1164
OAC5 1165
OAC6 1166
OAC7 1167
OAC8 1168
OAC9 1169
OACA 1170
OACB 1171
OACC 1172
OACD 1173
OACE 1174
OACF 1175
OAC0 1176
OAC1 1177
OAC2 1178
OAC3 1179
OAC4 1180
OAC5 1181
OAC6 1182
OAC7 1183
OAC8 1184
OAC9 1185
OACA 1186
OACB 1187
OACC 1188
OACD 1189
OACE 1190
OACF 1191
OAC0 1192
OAC1 1193
OAC2 1194
OAC3 1195
OAC4 1196
OAC5 1197
OAC6 1198
OAC7 1199
OAC8 1200
OAC9 1201
OACA 1202
OACB 1203
OACC 1204
OACD 1205
OACE 1206
OACF 1207
OAC0 1208
OAC1 1209
OAC2 1210
OAC3 1211
OAC4 1212
OAC5 1213
OAC6 1214
OAC7 1215
OAC8 1216
OAC9 1217
OACA 1218
OACB 1219
OACC 1220
OACD 1221
OACE 1222
OACF 1223
OAC0 1224
OAC1 1225
OAC2 1226
OAC3 1227
OAC4 1228
OAC5 1229
OAC6 1230
OAC7 1231
OAC8 1232
OAC9 1233
OACA 1234
OACB 1235
OACC 1236
OACD 1237
OACE 1238
OACF 1239
OAC0 1240
OAC1 1241
OAC2 1242
OAC3 1243
OAC4 1244
OAC5 1245
OAC6 1246
OAC7 1247
OAC8 1248
OAC9 1249
OACA 1250
OACB 1251
OACC 1252
OACD 1253
OACE 1254
OACF 1255
OAC0 1256
OAC1 1257
OAC2 1258
OAC3 1259
OAC4 1260
OAC5 1261
OAC6 1262
OAC7 1263
OAC8 1264
OAC9 1265
OACA 1266
OACB 1267
OACC 1268
OACD 1269
OACE 1270
OACF 1271
OAC0 1272
OAC1 1273
OAC2 1274
OAC3 1275
OAC4 1276
OAC5 1277
OAC6 1278
OAC7 1279
OAC8 1280
OAC9 1281
OACA 1282
OACB 1283
OACC 1284
OACD 1285
OACE 1286
OACF 1287
OAC0 1288
OAC1 1289
OAC2 1290
OAC3 1291
OAC4 1292
OAC5 1293
OAC6 1294
OAC7 1295
OAC8 1296
OAC9 1297
OACA 1298
OACB 1299
OACC 1300
OACD 1301
OACE 1302
OACF 1303
OAC0 1304
OAC1 1305
OAC2 1306
OAC3 1307
OAC4 1308
OAC5 1309
OAC6 1310
OAC7 1311
OAC8 1312
OAC9 1313
OACA 1314
OACB 1315
OACC 1316
OACD 1317
OACE 1318
OACF 1319
OAC0 1320
OAC1 1321
OAC2 1322
OAC3 1323
OAC4 1324
OAC5 1325
OAC6 1326
OAC7 1327
OAC8 1328
OAC9 1329
OACA 1330
OACB 1331
OACC 1332
OACD 1333
OACE 1334
OACF 1335
OAC0 1336
OAC1 1337
OAC2 1338
OAC3 1339
OAC4 1340
OAC5 1341
OAC6 1342
OAC7 1343
OAC8 1344
OAC9 1345
OACA 1346
OACB 1347
OACC 1348
OACD 1349
OACE 1350
OACF 1351
OAC0 1352
OAC1 1353
OAC2 1354
OAC3 1355
OAC4 1356
OAC5 1357
OAC6 1358
OAC7 1359
OAC8 1360
OAC9 1361
OACA 1362
OACB 1363
OACC 1364
OACD 1365
OACE 1366
OACF 1367
OAC0 1368
OAC1 1369
OAC2 1370
OAC3 1371
OAC4 1372
OAC5 1373
OAC6 1374
OAC7 1375
OAC8 1376
OAC9 1377
OACA 1378
OACB 1379
OACC 1380
OACD 1381
OACE 1382
OACF 1383
OAC0 1384
OAC1 1385
OAC2 1386
OAC3 1387
OAC4 1388
OAC5 1389
OAC6 1390
OAC7 1391
OAC8 1392
OAC9 1393
OACA 1394
OACB 1395
OACC 1396
OACD 1397
OACE 1398
OACF 1399
OAC0 1400
OAC1 1401
OAC2 1402
OAC3 1403
OAC4 1404
OAC5 1405
OAC6 1406
OAC7 1407
OAC8 1408
OAC9 1409
OACA 1410
OACB 1411
OACC 1412
OACD 1413
OACE 1414
OACF 1415
OAC0 1416
OAC1 1417
OAC2 1418
OAC3 1419
OAC4 1420
OAC5 1421
OAC6 1422
OAC7 1423
OAC8 1424
OAC9 1425
OACA 1426
OACB 1427
OACC 1428
OACD 1429
OACE 1430
OACF 1431
OAC0 1432
OAC1 1433
OAC2 1434
OAC3 1435
OAC4 1436
OAC5 1437
OAC6 1438
OAC7 1439
OAC8 1440
OAC9 1441
OACA 1442
OACB 1443
OACC 1444
OACD 1445
OACE 1446
OACF 1447
OAC0 1448
OAC1 1449
OAC2 1450
OAC3 1451
OAC4 1452
OAC5 1453
OAC6 1454
OAC7 1455
OAC8 1456
OAC9 1457
OACA 1458
OACB 1459
OACC 1460
OACD 1461
OACE 1462
OACF 1463
OAC0 1464
OAC1 1465
OAC2 1466
OAC3 1467
OAC4 1468
OAC5 1469
OAC6 1470
OAC7 1471
OAC8 1472
OAC9 1473
OACA 1474
OACB 1475
OACC 1476
OACD 1477
OACE 1478
OACF 1479
OAC0 1480
OAC1 1481
OAC2 1482
OAC3 1483
OAC4 1484
OAC5 1485
OAC6 1486
OAC7 1487
OAC8 1488
OAC9 1489
OACA 1490
OACB 1491
OACC 1492
OACD 1493
OACE 1494
OACF 1495
OAC0 1496
OAC1 1497
OAC2 1498
OAC3 1499
OAC4 1500
OAC5 1501
OAC6 1502
OAC7 1503
OAC8 1504
OAC9 1505
OACA 1506
OACB 1507
OACC 1508
OACD 1509
OACE 1510
OACF 1511
OAC0 1512
OAC1 1513
OAC2 1514
OAC3 1515
OAC4 1516
OAC5 1517
OAC6 1518
OAC7 1519
OAC8 1520
OAC9 1521
OACA 1522
OACB 1523
OACC 1524
OACD 1525
OACE 1526
OACF 1527
OAC0 1528
OAC1 1529
OAC2 1530
OAC3 1531
OAC4 1532
OAC5 1533
OAC6 1534
OAC7 1535
OAC8 1536
OAC9 1537
OACA 1538
OACB 1539
OACC 1540
OACD 1541
OACE 1542
OACF 1543
OAC0 1544
OAC1 1545
OAC2 1546
OAC3 1547
OAC4 1548
OAC5 1549
OAC6 1550
OAC7 1551
OAC8 1552
OAC9 1553
OACA 1554
OACB 1555
OACC 1556
OACD 1557
OACE 1558
OACF 1559
OAC0 1560
OAC1 1561
OAC2 1562
OAC3 1563
OAC4 1564
OAC5 1565
OAC6 1566
OAC7 1567
OAC8 1568
OAC9 1569
OACA 1570
OACB 1571
OACC 1572
OACD 1573
OACE 1574
OACF 1575
OAC0 1576
OAC1 1577
OAC2 1578
OAC3 1579
OAC4 1580
OAC5 1581
OAC6 1582
OAC7 1583
OAC8 1584
OAC9 1585
OACA 1586
OACB 1587
OACC 1588
OACD 1589
OACE 1590
OACF 1591
OAC0 1592
OAC1 1593
OAC2 1594
OAC3 1595
OAC4 1596
OAC5 1597
OAC6 1598
OAC7 1599
OAC8 1600
OAC9 1601
OACA 1602
OACB 1603
OACC 1604
OACD 1605
OACE 1606
OACF 1607
OAC0 1608
OAC1 1609
OAC2 1610
OAC3 1611
OAC4 1612
OAC5 1613
OAC6 1614
OAC7 1615
OAC8 1616
OAC9 1617
OACA 1618
OACB 1619
OACC 1620
OACD 1621
OACE 1622
OACF 1623
OAC0 1624
OAC1 1625
OAC2 1626
OAC3 1627
OAC4 1628
OAC5 1629
OAC6 1630
OAC7 1631
OAC8 1632
OAC9 1633
OACA 1634
OACB 1635
OACC 1636
OACD 1637
OACE 1638
OACF 1639
OAC0 1640
OAC1 1641
OAC2 1642
OAC3 1643
OAC4 1644
OAC5 1645
OAC6 1646
OAC7 1647
OAC8 1648
OAC9 1649
OACA 1650
OACB 1651
OACC 1652
OACD 1653
OACE 1654
OACF 1655
OAC0 1656
OAC1 1657
OAC2 1658
OAC3 1659
OAC4 1660
OAC5 1661
OAC6 1662
OAC7 1663
OAC8 1664
OAC9 1665
OACA 1666
OACB 1667
OACC 1668
OACD 1669
OACE 1670
OACF 1671
OAC0 1672
OAC1 1673
OAC2 1674
OAC3 1675
OAC4 1676
OAC5 1677
OAC6 1678
OAC7 1679
OAC8 1680
OAC9 1681
OACA 1682
OACB 1683
OACC 1684
OACD 1685
OACE 1686
OACF 1687
OAC0 1688
OAC1 1689
OAC2 1690
OAC3 1691
OAC4 1692
OAC5 1693
OAC6 1694
OAC7 1695
OAC8 1696
OAC9 1697
OACA 1698
OACB 1699
OACC 1700
OACD 1701
OACE 1702
OACF 1703
OAC0 1704
OAC1 1705
OAC2 1706
OAC3 1707
OAC4 1708
OAC5 1709
OAC6 1710
OAC7 1711
OAC8 1712
OAC9 1713
OACA 1714
OACB 1715
OACC 1716
OACD 1717
OACE 1718
OACF 1719
OAC0 1720
OAC1 1721
OAC2 1722
OAC3 1723
OAC4 1724
OAC5 1725
OAC6 1726
OAC7 1727
OAC8 1728
OAC9 1729
OACA 1730
OACB 1731
OACC 1732
OACD 1733
OACE 1734
OACF 1735
OAC0 1736
OAC1 1737
OAC2 1738
OAC3 1739
OAC4 1740
OAC5 1741
OAC6 1742
OAC7 1743
OAC8 1744
OAC9 1745
OACA 1746
OACB 1747
OACC 1748
OACD 1749
OACE 1750
OACF 1751
OAC0 1752
OAC1 1753
OAC2 1754
OAC3 1755
OAC4 1756
OAC5 1757
OAC6 1758
OAC7 1759
OAC8 1760
OAC9 1761
OACA 1762
OACB 1763
OACC 1764
OACD 1765
OACE 1766
OACF 1767
OAC0 1768
OAC1 1769
OAC2 1770
OAC3 1771
OAC4 1772
OAC5 1773
OAC6 1774
OAC7 1775
OAC8 1776
OAC9 1777
OACA 1778
OACB 1779
OACC 1780
OACD 1781
OACE 1782
OACF 1783
OAC0 1784
OAC1 1785
OAC2 1786
OAC3 1787
OAC4 1788
OAC5 1789
OAC6 1790
OAC7 1791
OAC8 1792
OAC9 1793
OACA 1794
OACB 1795
OACC 1796
OACD 1797
OACE 1798
OACF 1799
OAC0 1800
OAC1 1801
OAC2 1802
OAC3 1803
OAC4 1804
OAC5 1805
OAC6 1806
OAC7 1807
OAC8 1808
OAC9 1809
OACA 1810
OACB 1811
OACC 1812
OACD 1813
OACE 1814
OACF 1815
OAC0 1816
OAC1 1817
OAC2 1818
OAC3 1819
OAC4 1820
OAC5 1821
OAC6 1822
OAC7 1823
OAC8 1824
OAC9 1825
OACA 1826
OACB 1827
OACC 1828
OACD 1829
OACE 1830
OACF 1831
OAC0 1832
OAC1 1833
OAC2 1834
OAC3 1835
OAC4 1836
OAC5 1837
OAC6 1838
OAC7 1839
OAC8 1840
OAC9 1841
OACA 1842
OACB 1843
OACC 1844
OACD 1845
OACE 1846
OACF 1847
OAC0 1848
OAC1 1849
OAC2 1850
OAC3 1851
OAC4 1852
OAC5 1853
OAC6 1854
OAC7 1855
OAC8 1856
OAC9 1857
OACA 1858
OACB 1859
OACC 1860
OACD 1861
OACE 1862
OACF 1863
OAC0 1864
OAC1 1865
OAC2 1866
OAC3 1867
OAC4 1868
OAC5 1869
OAC6 1870
OAC7 1871
OAC8 1872
OAC9 1873
OACA 1874
OACB 1875
OACC 1876
OACD 1877
OACE 1878
OACF 1879
OAC0 1880
OAC1 1881
OAC2 1882
OAC3 1883
OAC4 1884
OAC5 1885
OAC6 1886
OAC7 1887
OAC8 1888
OAC9 1889
OACA 1890
OACB 1891
OACC 1892
OACD 1893
OACE 1894
OACF 1895
OAC0 1896
OAC1 1897
OAC2 1898
OAC3 1899
OAC4 1900
OAC5 1901
OAC6 1902
OAC7 1903
OAC8 1904
OAC9 1905
OACA 1906
OACB 1907
OACC 1908
OACD 1909
OACE 1910
OACF 1911
OAC0 1912
OAC1 1913
OAC2 1914
OAC3 1915
OAC4 1916
OAC5 1917
OAC6 1918
OAC7 1919
OAC8 1920
OAC9 1921
OACA 1922
OACB 1923
OACC 1924
OACD 1925
OACE 1926
OACF 1927
OAC0 1928
OAC1 1929
OAC2 1930
OAC3 1931
OAC4 1932
OAC5 1933
OAC6 1934
OAC7 1935
OAC8 1936
OAC9 1937
OACA 1938
OACB 1939
OACC 1940
OACD 1941
OACE 1942
OACF 1943
OAC0 1944
OAC1 1945
OAC2 1946
OAC3 1947
OAC4 1948
OAC5 1949
OAC6 1950
OAC7 1951
OAC8 1952
OAC9 1953
OACA 1954
OACB 1955
OACC 1956
OACD 1957
OACE 1958
OACF 1959
OAC0 1960
OAC1 1961
OAC2 1962
OAC3 1963
OAC4 1964
OAC5 1965
OAC6 1966
OAC7 1967
OAC8 1968
OAC9 1969
OACA 1970
OACB 1971
OACC 1972
OACD 1973
OACE 1974
OACF 1975
OAC0 1976
OAC1 1977
OAC2 1978
OAC3 1979
OAC4 1980
OAC5 1981
OAC6 1982
OAC7 1983
OAC8 1984
OAC9 1985
OACA 1986
OACB 1987
OACC 1988
OACD 1989
OACE 1990
OACF 1991
OAC0 1992
OAC1 1993
OAC2 1994
OAC3 1995
OAC4 1996
OAC5 1997
OAC6 1998
OAC7 1999
OAC8 2000
OAC9 2001
OACA 2002
OACB 2003
OACC 2004
OACD 2005
OACE 2006
OACF 2007
OAC0 2008
OAC1 2009
OAC2 2010
OAC3 2011
OAC4 2012
OAC5 2013
OAC6 2014
OAC7 2015
OAC8 2016
OAC9 2017
OACA 2018
OACB 2019
OACC 2020
OACD 2021
OACE 2022
OACF 2023
OAC0 2024
OAC1 2025
OAC2 2026
OAC3 2027
OAC4 2028
OAC5 2029
OAC6 2030
OAC7 2031
OAC8 2032
OAC9 2033
OACA 2034
OACB 2035
OACC 2036
OACD 2037
OACE 2038
OACF 2039
OAC0 2040
OAC1 2041
OAC2 2042
OAC3 2043
OAC4 2044
OAC5 2045
OAC6 2046
OAC7 2047
OAC8 2048
OAC9 2049
OACA 2050
OACB 2051
OACC 2052
OACD 2053
OACE 2054
OACF 2055
OAC0 2056
OAC1 2057
OAC2 2058
OAC3 2059
OAC4 2060
OAC5 2061
OAC6 2062
OAC7 2063
OAC8 2064
OAC9 2065
OACA 2066
OACB 2067
OACC 2068
OACD 2069
OACE 2070
OACF 2071
OAC0 2072
OAC1 2073
OAC2 2074
OAC3 2075
OAC4 2076
OAC5 2077
OAC6 2078
OAC7 2079
OAC8 2080
OAC9 2081
OACA 2082
OACB 2083
OACC 2084
OACD 2085
OACE 2086
OACF 2087
OAC0 2088
OAC1 2089
OAC2 2090
OAC3 2091
OAC4 2092
OAC5 2093
OAC6 2094
OAC7 2095
OAC8 2096
OAC9 2097
OACA 2098
OACB 2099
OACC 2100
OACD 2101
OACE 2102
OACF 2103
OAC0 2104
OAC1 2105
OAC2 2106
OAC3 2107
OAC4 2108
OAC5 2109
OAC6 2110
OAC7 2111
OAC8 2112
OAC9 2113
OACA 2114
OACB 2115
OACC 2116
OACD 2117
OACE 2118
OACF 2119
OAC0 2120
OAC1 2121
OAC2 2122
OAC3 2123
OAC4 2124
OAC5 2125
OAC6 2126
OAC7 2127
OAC8 2128
OAC9 2129
OACA 2130
OACB 2131
OACC 2132
OACD 2133
OACE 2134
OACF 2135
OAC0 2136
OAC1 2137
OAC2 2138
OAC3 2139
OAC4 2140
OAC5 2141
OAC6 2142
OAC7 2143
OAC8 2144
OAC9 2145
OACA 2146
OACB 2147
OACC 2148
OACD 2149
OACE 2150
OACF 2151
OAC0 2152
OAC1 2153
OAC2 2154
OAC3 2155
OAC4 2156
OAC5 2157
OAC6 2158
OAC7 2159
OAC8 2160
OAC9 2161
OACA 2162
OACB 2163
OACC 2164
OACD 2165
OACE 2166
OACF 2167
OAC0 2168
OAC1 2169
OAC2 2170
OAC3 2171
OAC4 2172
OAC5 2173
OAC6 2174
OAC7 2175
OAC8 2176
OAC9 2177
OACA 2178
OACB 2179
OACC 2180
OACD 2181
OACE 2182
OACF 2183
OAC0 2184
OAC1 2185
OAC2 2186
OAC3 2187
OAC4 2188
OAC5 2189
OAC6 2190
OAC7 2191
OAC8 2192
OAC9 2193
OACA 2194
OACB 2195
OACC 2196
OACD 2197
OACE 2198
OACF 2199
OAC0 2200
OAC1 2201
OAC2 2202
OAC3 2203
OAC4 2204
OAC5 2205
OAC6 2206
OAC7 2207
OAC8 2208
OAC9 2209
OACA 2210
OACB 2211
OACC 2212
OACD 2213
OACE 2214
OACF 2215
OAC0 2216
OAC1 2217
OAC2 2218
OAC3 2219
OAC4 2220
OAC5 2221
OAC6 2222
OAC7 2223
OAC8 2224
OAC9 2225
OACA 2226
OACB 2227
OACC 2228
OACD 2229
OACE 2230
OACF 2231
OAC0 2232
OAC1 2233
OAC2 2234
OAC3 2235
OAC4 2236
OAC5 2237
OAC6 2238
OAC7 2239
OAC8 2240
OAC9 2241
OACA 2242
OACB 2243
OACC 2244
OACD 2245
OACE 2246
OACF 2247
OAC0 2248
OAC1 2249
OAC2 2250
OAC3 2251
OAC4 2252
OAC5 2253
OAC6 2254
OAC7 2255
OAC8 2256
OAC9 2257
OACA 2258
OACB 2259
OACC 2260
OACD 2261
OACE 2262
OACF 2263
OAC0 2264
OAC1 2265
OAC2 2266
OAC3 2267
OAC4 2268
OAC5 2269
OAC6 2270
OAC7 2271
OAC8 2272
OAC9 2273
OACA 2274
OACB 2275
OACC 2276
OACD 2277
OACE 2278
OACF 2279
OAC0 2280
OAC1 2281
OAC2 2282
OAC3 2283
OAC4 2284
OAC5 2285
OAC6 2286
OAC7 2287
OAC8 2288
OAC9 2289
OACA 2290
OACB 2291
OACC 2292
OACD 2293
OACE 2294
OACF 2295
OAC0 2296
OAC1 2297
OAC2 2298
OAC3 2299
OAC4 2300
OAC5 2301
OAC
```

SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00 Page 26
GETJPI TESTS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1 (1)

			OBCC	967	TEST_END	
	004C'CF	DD	OBCC		PUSHL	W^TMD_ADDR
	0048'CF	DD	OBDO		PUSHL	W^TMN_ADDR
		DD	OBDA		PUSHL	#2
	0044'CF	DD	OBDA		PUSHL	W^MOD_MSG_CODE
	00000000'GF	FB	OBDA		CALLS	\$\$\$T1,G^LIBSSIGNAL
0044'CF	01	1C	OBE1		INSV	#1,\$\$SSV_INHIB_MSG,#1,W^MOD_MSG_CODE
	0044'CF	DD	OBE8		PUSHL	W^MOD_MSG_CODE
00000000'GF	01	FB	OBE8		CALLS	#1,G^SYS\$EXIT

```
0008'CF 14 AD 28 28 04 00FC
0008'CF 14 AD 28 29 13 00FC
56 53 00000008'8F 56 04 C6 0C24 1023
7E 56 02 81 0C27 1024
51 03 CA 0C2B 1025

0BF3 969 .SBTTL ROUTINES
0BF3 970 .SBTTL REG_SAVE
0BF3 971 :++
0BF3 972 : FUNCTIONAL DESCRIPTION:
0BF3 973 : Subroutine to save R2-R11 in the register save location.
0BF3 974 :
0BF3 975 : CALLING SEQUENCE:
0BF3 976 : PUSHL #0 ; save a dummy parameter
0BF3 977 : CALLS #1,W^REG_SAVE ; save R2-R11
0BF3 978 :
0BF3 979 : INPUT PARAMETERS:
0BF3 980 : NONE
0BF3 981 :
0BF3 982 : OUTPUT PARAMETERS:
0BF3 983 : NONE
0BF3 984 :
0BF3 985 :--
0BF3 986 :
0BF3 987 REG_SAVE:
0BF3 988 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0BF3 989 MOVC3 #4*10,^X14(FP),W^REG_SAVE_AREA ; save the registers in the program
0BF3 990 RET
0BF3 991 .SBTTL REG_CHECK
0BF3 992 :++
0BF3 993 : FUNCTIONAL DESCRIPTION:
0BF3 994 : Subroutine to test R0 & R2-R11 for proper content after a service
0BF3 995 : execution. A snapshot is taken by the REG_SAVE routine at the
0BF3 996 : beginning of each step and this routine is executed after the
0BF3 997 : services have been executed.
0BF3 998 :
0BF3 999 : CALLING SEQUENCE:
0BF3 1000 : PUSHL #SS$ XXXXXX ; push expected R0 contents
0BF3 1001 : CALLS #1,W^REG_CHECK ; execute this routine
0BF3 1002 :
0BF3 1003 : INPUT PARAMETERS:
0BF3 1004 : expected R0 contents on the stack
0BF3 1005 :
0BF3 1006 : OUTPUT PARAMETERS:
0BF3 1007 : possible error messages printed using $PUTMSG
0BF3 1008 :
0BF3 1009 :--
0BF3 1010 :
0BF3 1011 REG_CHECK:
0BF3 1012 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0BF3 1013 CMPL 4(AP),R0 ; is this the right fail code?
0BF3 1014 BEQL 10$ ; br if yes
0BF3 1015 PUSHL R0 ; push received data
0BF3 1016 PUSHL 4(AP) ; push expected data
0BF3 1017 PUSHAL W^EXP ; push the string variable
0BF3 1018 CALLS #3,W^PRINT_FAIL ; print the error message
0BF3 1019 10$:
0BF3 1020 CMPC3 #4*10,^X14(FP),W^REG_SAVE_AREA ; check all but R0
0BF3 1021 BEQL 20$ ; br if O.K.
0BF3 1022 SUBL3 #REG_SAVE_AREA,R3,R6 ; calculate the register number
0BF3 1023 DIVL2 #4,R6
0BF3 1024 ADDB3 #^X2,R6,-(SP) ; set number past R0-R1 and save
0BF3 1025 BICL2 #3,R1 ; backup to register boundrys
```

```
53 03 CA 0C2E 1026 BICL2 #3,R3
61 DD 0C31 1027 PUSHL (R1)
63 DD 0C33 1028 PUSHL (R3)
006D'CF DF 0C35 1029 PUSHAL W^REG
0C3F'CF 04 FB 0C39 1030 CALLS #4,W^PRINT_FAIL
04 0C3E 1031 20$: RET
0C3E 1032 .SBTTL PRINT_FAIL
0C3F 1033
0C3F 1034
0C3F 1035 :++
0C3F 1036 : FUNCTIONAL DESCRIPTION:
0C3F 1037 : Subroutine to report failures using $PUTMSG
0C3F 1038 : CALLING SEQUENCE:
0C3F 1039 : Mode #1 PUSHL EXPECTED Mode #2 PUSHL REG NUMBER
0C3F 1040 : PUSHL RECEIVED PUSHL EXPECTED
0C3F 1041 : PUSHAL STRING VAR PUSHL RECEIVED
0C3F 1042 : CALLS #3,W^PRINT_FAIL PUSHAL STRING VAR
0C3F 1043 : CALLS #4,W^PRINT_FAIL
0C3F 1044 : Mode #3 PUSHAL STRING VAR
0C3F 1045 : CALLS #1,W^PRINT_FAIL
0C3F 1046
0C3F 1047 : INPUT PARAMETERS:
0C3F 1048 : listed above
0C3F 1049
0C3F 1050 : OUTPUT PARAMETERS:
0C3F 1051 : an error message is printed using $PUTMSG
0C3F 1052
0C3F 1053 :--
0C3F 1054
0C3F 1055 PRINT_FAIL:
003C 0C3F 1056 .WORD ^M<R2,R3,R4,R5>
0C41 1057 $FAO_S W^CS1,W^MESSAGEL,W^MSGL,#TEST_MOD_NAME,W^SERV_NAME,W^CURRENT_TC
0C62 1058 $PUTMSG_S W^MSGVEC
04 6C 91 0C73 1059 CMPB (AP),#4
26 13 0C76 1060 BEQL 10$
01 6C 91 0C78 1061 CMPB (AP),#1
48 13 0C7B 1062 BEQL 20$
0C7D 1063 $FAO_S W^CS2,W^MESSAGEL,W^MSGL,4(AP),8(AP),4(AP),12(AP)
40 11 0C9C 1064 BRB 30$
0C9E 1065 10$: $FAO_S W^CS3,W^MESSAGEL,W^MSGL,4(AP),16(AP),8(AP),4(AP),16(AP),12(AP)
19 11 0C9E 1066 BRB 30$
0CC3 1067 20$: $FAO_S W^MSGVEC1+12
022F'CF 04 AC D0 0CC5 1068 MOVL 4(AP),W^MSGVEC1+12
0CCB 1070 $PUTMSG_S W^MSGVEC1
11 11 0CDC 1071 BRB 40$
0CDE 1072 30$: $PUTMSG_S W^MSGVEC
0CDE 1073 40$: $PUTMSG_S W^MSGVEC
0CEF 1074 CALLS #0,W^MODE_ID
0003'CF 00 FB 0CEF 1075 MOVAL W^TEST_MOD_FAIL,W^TMD_ADDR
004C'CF 002A'CF DE 0CF4 1076 INSV #ERROR,#0,#3,W^MOD_MSG_CODE
0044'CF 03 00 02 F0 0CFB 1077 RET
04 0D02 1078
```

```
0003 1080 .SBTTL MODE_ID
0003 1081 :++
0003 1082 : FUNCTIONAL DESCRIPTION:
0003 1083 : Subroutine to identify the mode that an exit handler is in.
0003 1084 :
0003 1085 : CALLING SEQUENCE:
0003 1086 : CALLS #0,W^MODE_ID
0003 1087 :
0003 1088 : INPUT PARAMETERS:
0003 1089 : MODE contains an address pointing to an ascii string desc.
0003 1090 : of the current CPU mode.
0003 1091 :
0003 1092 : OUTPUT PARAMETERS:
0003 1093 : NONE
0003 1094 :
0003 1095 :--
0003 1096 :
0003 1097 MODE_ID:
003C 0003 1098 .WORD ^M<R2,R3,R4,R5>
0003 1099 $FAO S W^CS5,W^MESSAGEL,W^MSGL,MODE ; format the error message
04 001E 1100 $PUTMSG_S W^MSGVEC ; print the mode message
002F 1101 RET
0030 1102 .SBTTL CRE_CHECK
0030 1103 :++
0030 1104 : FUNCTIONAL DESCRIPTION:
0030 1105 : Routine to check the process exit status of a created process.
0030 1106 :
0030 1107 : CALLING SEQUENCE:
0030 1108 : CALLS #0,W^CRE_CHECK ; save R2-R11
0030 1109 :
0030 1110 : INPUT PARAMETERS:
0030 1111 : R6 = Expected process exit status
0030 1112 : R7 = PID check flag BIT0 = 1 means check the PID
0030 1113 :
0030 1114 : OUTPUT PARAMETERS:
0030 1115 : NONE
0030 1116 :
0030 1117 :--
0030 1118 :
0030 1119 CRE_CHECK:
OFFC 0030 1120 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0032 1121 $QIOW_S
0032 1122 EFN = #1,-
0032 1123 FUNC = #10$, READVBLK,-
0032 1124 CHAN = W^MBCHAN,-
0032 1125 IOSB = W^IOSTATUS,-
0032 1126 P1 = W^MBUF,-
0032 1127 P2 = #100
56 024F'CF D1 0D59 1127 CMPL W^MBUF+ACC$$_FINALSTS,R6 ; read the mail
0F 13 0D5E 1128 BEQL 10$ ; is the status as expected?
024F'CF DD 0D60 1129 PUSHL W^MBUF+ACC$$_FINALSTS ; br if OK
56 DD 0D64 1130 PUSHL R6 ; push received
00D8'CF DF 0D66 1131 PUSHAL W^EXP ; push expected
FEDO CF 03 FB 0D6A 1132 CALLS #3,W^PRINT_FAIL ; push string variable
0D6F 1133 10$: ; print the failure
0D6F 1134 BLBC R7,20$ ; should we check the PID?
0247'CF 1A 57 E9 0D6F 1135 CMPL W^PID1,W^IOSTATUS+4 ; check the PID
023B'CF D1 0D72 1135 BEQL 20$ ; br if its good
11 13 0D79 1136
```

```
0247'CF DD OD7B 1137 PUSHL W^IOSTATUS+4 ; push received
023B'CF DD OD7F 1138 PUSHL W^PID1 ; push expected
0163'CF DF OD83 1139 PUSHAL W^PID_STR ; push the string variable
FEB3 CF 03 FB OD87 1140 CALLS #3,W^PRINT_FAIL ; print the failure
                                20$:
                                04 OD8C 1141 RET
                                OD8C 1142 .SBTTL JPI_CHECK
                                OD8D 1143
                                OD8D 1144 :++
                                OD8D 1145 : FUNCTIONAL DESCRIPTION:
                                OD8D 1146 : Subroutine to check the results of a JPI service
                                OD8D 1147 :
                                OD8D 1148 : CALLING SEQUENCE:
                                OD8D 1149 : CALLS #0,W^JPI_CHECK ; check the results
                                OD8D 1150 :
                                OD8D 1151 : INPUT PARAMETERS:
                                OD8D 1152 : R6 = questionable data address
                                OD8D 1153 : R7 = good data address
                                OD8D 1154 : R8 = byte count
                                OD8D 1155 :
                                OD8D 1156 : OUTPUT PARAMETERS:
                                OD8D 1157 : NONE
                                OD8D 1158 :
                                OD8D 1159 : --
                                OD8D 1160
                                00000D99 OD8D 1161 ARGST1:
                                OD8D 1162 .BLKL 3
                                OD99 1163 CTRSTR:
                                OD99 1164 .ASCID /data error at offset !XW, good data = !XB bad data = !XB./
65 20 61 74 61 64 00000DA1'010E0000'
73 66 66 6F 20 74 61 20 72 6F 72 72 ODA7
64 6F 6F 67 20 2C 57 58 21 20 74 65 OD83
20 42 58 21 20 3D 20 61 74 61 64 20 ODBF
21 20 3D 20 61 74 61 64 20 64 61 62 ODCB
                                2E 42 58 ODD7
                                ODDA 1165 :
                                ODDA 1166 JPI_CHECK:
                                ODDA 1167 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                ODDC 1168 CMPC3 R8,(R6),(R7) ; check the buffer
                                ODE0 1169 BEQL 10$ ; br if good
                                ODE2 1170 SUBL3 #ACCOUNT,R1,W^ARGST1 ; get buffer offset
                                ODEC 1171 MOVZBL (R3),W^ARGST1+4 ; get the good data
                                ODF1 1172 MOVZBL (R1),W^ARGST1+8 ; get the bad data
                                ODF6 1173 $FAOL S W^CTRSTR,W^ML,W^GETBUF,W^ARGST1 ; make it readable
                                FE29 CF 01 DF OE0D 1174 PUSHAL W^ML ; push the desc. address
                                FE29 CF 01 FB OE11 1175 CALLS #1,W^PRINT_FAIL ; print the failure
                                OE16 1176 10$:
                                04 OE16 1177 RET
```

```
0E17 1180 MOD_MSG_PRINT:
0E17 1181 :
0E17 1182 : *****
0E17 1183 : *
0E17 1184 : * PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES
0E17 1185 : * (USING THE PUTMSG MACRO).
0E17 1186 : *
0E17 1187 : *****
0E17 1188 :
05 0E17 1189 PUTMSG <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> : PRINT MSG
0E32 1190 RSB ; ... AND RETURN TO CALLER
0E33 1191 :
0E33 1192 CHMRTN:
0E33 1193 : *****
0E33 1194 : *
0E33 1195 : * CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER
0E33 1196 : * A CMKRN, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED
0E33 1197 : * BY THE MODE MACRO ('TO' OPTION). IT MERELY DOES
0E33 1198 : * A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS
0E33 1199 : * THE EFFECT OF RETURNING TO THE END OF THE MODE
0E33 1200 : * MACRO EXPANSION.
0E33 1201 : *
0E33 1202 : *****
0E33 1203 :
00000059'FF 0000 0E33 1204 .WORD 0 ; ENTRY MASK
17 0E35 1205 JMP @CHM_CONT ; RETURN TO MODE MACRO IN NEW MODE
0E3B 1206 :
0E3B 1207 : * RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
0E3B 1208 :
0E3B 1209 .END SATSSS35
```

SATSSS35
Symbol table

I 10
- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC] SATSSS35.MAR;1

Page 32
(3)

SSARGS	= 00000007		
SS1	= 00000004		
SS2	= 00000004		
ACCSL_FINALSTS	= 00000004		
ACCOUNT	00000000	R	02
ACCOUNTL	00000008	R	02
APTCNT	00000050	R	02
APTCNTL	00000054	R	02
ARGLST1	00000080	R	05
ASTACT	00000056	R	02
ASTACTL	0000005A	R	02
ASTCNT	00000062	R	02
ASTCNTL	00000066	R	02
ASTEN	0000005C	R	02
ASTENL	00000060	R	02
ASTLM	00000068	R	02
ASTLML	0000006C	R	02
AST_PARAM	000000E6	R	03
AUTHPRIV	0000006E	R	02
AUTHPRIVL	00000076	R	02
BIOCNT	00000078	R	02
BIOCNTL	0000007C	R	02
BIOLM	0000007E	R	02
BIOLML	00000082	R	02
BP	000000F8	R	03
BUF	00000133	R	04
BUF10	00000084	R	02
BUF10L	00000088	R	02
BYTCNT	0000008A	R	02
BYTCNTL	0000008E	R	02
BYTLM	00000090	R	02
BYTLM	00000094	R	02
CHMRTN	00000E33	R	05
CHM_CONT	00000059	R	04
CPULIM	0000000A	R	02
CPULIML	0000000E	R	02
CPUTIM	00000096	R	02
CPUTIML	0000009A	R	02
CRE	0000008B	R	04
CREPRC	0000016E	R	03
CREPRCS_BASPRI	= 00000024		
CREPRCS_ERROR	= 00000014		
CREPRCS_IMAGE	= 00000008		
CREPRCS_INPUT	= 0000000C		
CREPRCS_ITMLST	= 00000034		
CREPRCS_MBXUNT	= 0000002C		
CREPRCS_NARGS	= 0000000D		
CREPRCS_OUTPUT	= 00000010		
CREPRCS_PIDADR	= 00000004		
CREPRCS_PRCNAM	= 00000020		
CREPRCS_PRIVADR	= 00000018		
CREPRCS_QUOTA	= 0000001C		
CREPRCS_STSFLG	= 00000030		
CREPRCS_UIC	= 00000028		
CRE_CHECK	000000D3	R	05
CS1	00000031	R	03
CS2	00000063	R	03

CS3	00000090	R	03
CS5	000000C3	R	03
CTLSGL_PHD	*****	X	05
CTRSTR	00000099	R	05
CURPRIV	00000010	R	02
CURPRIVL	00000018	R	02
CURRENT_TC	00000004	R	04
DFPFC	0000009C	R	02
DFPFC	000000A0	R	02
DFWSCNT	000000A2	R	02
DFWSCNTL	000000A6	R	02
DIBSW_UNIT	= 0000000C		
DIOCNT	000000A8	R	02
DIOCNTL	000000AC	R	02
DIOLM	000000AE	R	02
DIOLML	000000B2	R	02
DIRIO	000000B4	R	02
DIRIOL	000000B8	R	02
DIRTY	00000266	R	03
EFCS	000000BA	R	02
EFCSL	000000BE	R	02
EFCL	000000C0	R	02
EFCL	000000C4	R	02
EFCL_NAME	0000014F	R	03
EFWM	000000C6	R	02
EFWML	000000CA	R	02
ERR	00000529	R	03
ERROR	= 00000002		
EXCVEC	000000CC	R	02
EXCVECL	000000D0	R	02
EXP	000000D8	R	03
FILCNT	000000D8	R	02
FILCNTL	000000DC	R	02
FILLM	000000DE	R	02
FILLML	000000E2	R	02
FINALEXC	000000D2	R	02
FINALEXCL	000000D6	R	02
FREPOVA	000000E4	R	02
FREPOVAL	000000E8	R	02
FREP1VA	000000EA	R	02
FREP1VAL	000000EE	R	02
GET	000000C3	R	04
GET1	000000E3	R	04
GETBUF	0000018B	R	04
GETJPI	00000175	R	03
GETJPIS_ASTADR	= 00000018		
GETJPIS_ASTPRM	= 0000001C		
GETJPIS_EFN	= 00000004		
GETJPIS_IOSB	= 00000014		
GETJPIS_ITMLST	= 00000010		
GETJPIS_NARGS	= 00000007		
GETJPIS_PIDADR	= 00000008		
GETJPIS_PRCNAM	= 0000000C		
GET_LIST	000001EE	R	03
GPGCNT	000000F0	R	02
GPGCNTL	000000F4	R	02
GRP	0000001A	R	02

SATSSS35
Symbol table

J 10
- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 33
(3)

GRPL	0000001E	R	02	JPIS_STATE	=	00000306		
IMAGE_NAME	0000053A	R	03	JPIS_STS	=	00000305		
IMAGNAME	000000F6	R	02	JPIS-TMBU	=	00000308		
IMAGNAMEL	00000176	R	02	JPIS-TQCNT	=	00000315		
IMAGPRIV	00000020	R	02	JPIS-TQLM	=	00000410		
IMAGPRIVL	00000028	R	02	JPIS-UIC	=	00000304		
IN	00000506	R	03	JPIS-USERNAME	=	00000202		
IOS_READVBLK	*****	X	05	JPIS-VIRTPEAK	=	00000200		
IOSTAT	00000233	R	04	JPIS-VOLUMES	=	00000205		
IOSTATUS	00000243	R	04	JPIS-WSAUTH	=	00000401		
ITEM_LIST	00000103	R	04	JPIS-WSPEAK	=	00000201		
JPIS_ACCOUNT	= 00000203			JPIS-WSQUOTA	=	00000402		
JPIS_APTCNT	= 0000030A			JPIS-WSSIZE	=	00000411		
JPIS_ASTACT	= 00000300			JPI_CHECK	00000DDA	R	05	
JPIS_ASTCNT	= 0000030E			JPI_GOOD	000004B6	R	03	
JPIS_ASTEN	= 00000301			JPI_GOOD_SHRT	000004C0	R	03	
JPIS_ASTLM	= 00000409			JPI_LIST_SIZE	= 00000044			
JPIS_AUTHPRIV	= 00000412			JPI_LIST_SIZE1	= 0000003A			
JPIS_BIOCNT	= 0000030F			JPI_PRV_MASK	= 1070BFEF			
JPIS_BIOLM	= 00000310			JPI_PRV_NMASK	= EF8F4010			
JPIS_BUFIO	= 0000040C			JPI_STS_MASK	= 0038C600			
JPIS_BYTCNT	= 00000311			JPI_STS_NMASK	= FFC739FF			
JPIS_BYTLM	= 0000031A			LIB\$SIGNAL	*****	X	05	
JPIS_CPULIM	= 0000040D			LOGINTIM	00000178	R	02	
JPIS_CPUTIM	= 00000407			LOGINTIML	0000017C	R	02	
JPIS_CURPRIV	= 00000400			MBCHAN	0000023F	R	04	
JPIS_DFPFC	= 00000406			MBNAM	00000188	R	03	
JPIS_DFWSCNT	= 00000403			MBUF	00000248	R	04	
JPIS_DIOCNT	= 00000312			MBXUN	00000241	R	04	
JPIS_DIOLM	= 00000313			MEM	0000002A	R	02	
JPIS_DIRIO	= 0000040B			MEML	0000002E	R	02	
JPIS_EFCS	= 00000317			MESSAGEL	00000217	R	04	
JPIS_EFCU	= 00000318			ML	00000183	R	04	
JPIS_EFWM	= 00000316			MODE	00000069	R	04	
JPIS_EXCVEC	= 00000100			MODE_ID	00000D03	R	05	
JPIS_FILCNT	= 00000314			MOD_MSG_CODE	00000044	R	04	
JPIS_FILLM	= 0000040F			MOD_MSG_PRINT	00000E17	R	05	
JPIS_FINALEXC	= 00000101			MSGC	00000083	R	04	
JPIS_FREPOVA	= 00000404			MSGVEC	000001A6	R	03	
JPIS_FREPIVA	= 00000405			MSGVEC1	00000223	R	04	
JPIS_GPGCNT	= 0000030C			NPRVMASK	0000019E	R	03	
JPIS_GRP	= 00000308			OUT	00000517	R	03	
JPIS-IMAGNAME	= 00000207			OWNER	0000017E	R	02	
JPIS-IMAGPRIV	= 00000413			OWNERL	00000182	R	02	
JPIS-LOGINTIM	= 00000206			PAGEFLTS	00000184	R	02	
JPIS-MEM	= 00000307			PAGEFLTSL	00000188	R	02	
JPIS-OWNER	= 00000303			PCBSV_BATCH	= 0000000E			
JPIS-PAGEFLTS	= 0000040A			PCBSV-HIBER	= 00000013			
JPIS-PGFLQUOTA	= 0000040E			PCBSV-LOGIN	= 00000014			
JPIS-PID	= 00000319			PCBSV-NETWRK	= 00000015			
JPIS-PPGCNT	= 0000030D			PCBSV-NOACNT	= 0000000F			
JPIS-PRCCNT	= 0000031B			PCBSV-SSFEXCU	= 00000009			
JPIS-PRCLM	= 00000408			PCBSV-SSRWAIT	= 0000000A			
JPIS-PRCNAM	= 0000031C			PGFLQUOTA	0000018A	R	02	
JPIS-PRI	= 00000302			PGFLQUOTAL	0000018E	R	02	
JPIS-PRIB	= 00000309			PHDSQ_PRIVMSK	= 00000000			
JPIS-PROCPRIV	= 00000204			PID	00000190	R	02	

SATSSS35
Symbol table

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) K 10
16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 34
(3)

PID1	0000023B	R	04
PIDL	00000194	R R	02
PID_STR	00000163	R R	03
PNS	00000109	R R	03
PPGCNT	00000196	R R	02
PPGCNTL	0000019A	R	02
PQLS_ASTLM	= 00000001		
PQLS_BIOLM	= 00000002		
PQLS_BYTLM	= 00000003		
PQLS_CPULM	= 00000004		
PQLS_DIOLM	= 00000005		
PQLS_FILLM	= 00000006		
PQLS_LISTEND	= 00000000		
PQLS_PGFLQUOTA	= 00000007		
PQLS_PRCLM	= 00000008		
PQLS_TQELM	= 00000009		
PQLS_WSDEFAULT	= 0000000B		
PQLS_WSQUOTA	= 0000000A		
PRCCNT	0000019C	R R	02
PRCCNTL	000001A0	R R	02
PRCLM	00000030	R R	02
PRCLML	00000034	R R	02
PRCNAM	000001A2	R R	02
PRCNAML	000001B1	R R	02
PRI	000001BD	R R	02
PRIB	000001C3	R R	02
PRIBL	000001C7	R R	02
PRIL	000001C1	R R	02
PRINT_FAIL	00000C3F	R R	05
PRIVMASK	00000051	R R	04
PRIVS	000002B3	R	04
PRIV_ARGS	= 00000002		
PROCPRIV	000001B3	R R	02
PROCPRIVL	000001BB	R R	02
PROC_NAME	0000054E	R R	03
PROC_UIC	00000565	R	03
PRVSV_CMEXEC	= 00000001		
PRVSV_CMKRNL	= 00000000		
PRVSV_DETACH	= 00000005		
PRVSV_DIAGNOSE	= 00000006		
PRVSV_GROUP	= 00000008		
PRVSV_GRPNAM	= 00000003		
PRVSV_LOG_IO	= 00000007		
PRVSV_NETMBX	= 00000014		
PRVSV_NOACNT	= 00000009		
PRVSV_PHY_IO	= 00000016		
PRVSV_PRMCB	= 0000000A		
PRVSV_PRMBX	= 0000000B		
PRVSV_PSWAPM	= 0000000C		
PRVSV_SETPRI	= 0000000D		
PRVSV_SETPRV	= 0000000E		
PRVSV_SYSNAM	= 00000002		
PRVSV_SYSPRV	= 0000001C		
PRVSV_TMPMBX	= 0000000F		
PRVSV_VOLPRO	= 00000015		
PRVMASK	00000196	R	03
PRVPRT	00000050	R	04

QUOTA_LIST	000001B6	R	03
REG	0000006D	R	04
REGNUM	0000007F	R	04
REG_CHECK	00000BFD	R	05
REG_SAVE	00000BF3	R	05
REG_SAVE_AREA	00000008	R	04
RETADR	0000005D	R	04
RMS\$ FNF	*****	X	05
SATSSS35	00000000	RG	05
SERV_NAME	0000021F	R	04
SHORT_LIST	000001FA	R	03
SHR\$ ABEND	= 000010E0		
SHR\$ BEGIND	= 00001038		
SHR\$ ENDEDD	= 00001080		
SHR\$ TEXT	= 00001130		
SS\$ NORMAL	*****	X	05
STATE	000001C9	R	02
STATEL	000001CD	R R	02
STATUS	00000065	R	04
STEP	= 00000016		
STP0	0000003D	R	05
STP1	000000BE	R	05
STP10	0000049C	R R	05
STP11	00000543	R R	05
STP12	000006CC	R R	05
STP13	0000075E	R R	05
STP14	000007FF	R R	05
STP15	00000824	R R	05
STP16	000008CF	R R	05
STP17	00000962	R R	05
STP18	0000099E	R R	05
STP19	00000A0A	R R	05
STP2	000000F3	R R	05
STP20	00000A4A	R R	05
STP21	00000AC1	R R	05
STP22	00000B48	R R	05
STP3	00000146	R R	05
STP4	0000018D	R R	05
STP5	000001E4	R R	05
STP6	00000224	R R	05
STP7	0000027F	R R	05
STP8	00000309	R R	05
STP9	000003AA	R R	05
STS	000001CF	R	02
STSSV_FAC NO	= 00000010		
STSSV_INHIB_MSG	= 0000001C		
STSFLGS	00000134	R	03
STSL	000001D3	R	02
SUCCESS	= 00000001		
SYSSASCEFC	*****	GX	05
SYSSCMKRNL	*****	GX	05
SYSSCREMBX	*****	GX	05
SYSSCREPRC	*****	GX	05
SYSSDACEFC	*****	GX	05
SYSSDELPRC	*****	GX	05
SYSEXIT	*****	GX	05
SYSSFAO	*****	X	05

SATSSS35
Symbol table

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) L 10
16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 35
(3)

SYSS\$FAOL	*****	GX	05
SYSS\$GETCHN	*****	GX	05
SYSS\$GETJPI	*****	GX	05
SYSS\$HIBER	*****	GX	05
SYSS\$PUTMSG	*****	GX	05
SYSS\$QIOW	*****	GX	05
SYSS\$SETAST	*****	GX	05
SYSS\$SETPRN	*****	GX	05
SYSS\$SETPRV	*****	GX	05
SYSS\$WAITFR	*****	GX	05
SYSS\$WAKE	*****	GX	05
TEST_MOD_BEGIN	00000019	R	03
TEST_MOD_FAIL	0000002A	R	03
TEST_MOD_NAME	00000000	R	03
TEST_MOD_NAME_D	00000009	R	03
TEST_MOD_SUCC	0000001F	R	03
TEST_PID	000002AF	R	04
TMBU	000001D5	R	02
TMBUL	000001D9	R	02
TMD_ADDR	0000004C	R	04
TMN_ADDR	00000048	R	04
TPID	00000000	R	04
TQCNT	000001DB	R	02
TQCNTL	000001DF	R	02
TQLM	00000036	R	02
TQLML	0000003A	R	02
UETP	= 00740000		
UETPS_ABEND	= 007410E0		
UETPS_BEGIN	= 00741038		
UETPS_ENDED	= 00741080		
UETPS_SATSMS	= 007480D9		
UETPS_TEXT	= 00741130		
UIC	0000003C	R	02
UICL	00000040	R	02
UIC_MSG	00000144	R	03
UM	0000017C	R	03
USERNAME	00000042	R	02
USERNAMEL	0000004E	R	02
VIRTPEAK	000001E7	R	02
VIRTPEAKL	000001EB	R	02
VOLUMES	000001E1	R	02
VOLUMESL	000001E5	R	02
WSAUTH	000001ED	R	02
WSAUTHL	000001F1	R	02
WSPEAK	000001F9	R	02
WSPEAKL	000001FD	R	02
WSQUOTA	000001F3	R	02
WSQUOTAL	000001F7	R	02
WSSIZE	000001FF	R	02
WSSIZEL	00000203	R	02

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
ITEM_LIST	00000205 (517.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
RODATA	00000569 (1385.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	00000288 (699.)	04 (4.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS35	00000E3B (3643.)	05 (5.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.08	00:00:00.31
Command processing	134	00:00:00.63	00:00:02.15
Pass 1	472	00:00:19.08	00:00:32.37
Symbol table sort	0	00:00:01.51	00:00:01.52
Pass 2	257	00:00:04.93	00:00:08.65
Symbol table output	46	00:00:00.38	00:00:00.54
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	949	00:00:26.66	00:00:45.59

The working set limit was 2000 pages.
116238 bytes (228 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1010 non-local and 25 local symbols.
1209 source lines were read in Pass 1, producing 42 object records in Pass 2.
63 pages of virtual memory were used to define 57 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	10
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	41
TOTALS (all libraries)	53

1207 GETS were required to define 53 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSSS35/OBJ=OBJ\$:SATSSS35 MSRC\$:SATSSS35/UPDATE=(ENH\$:SATSSS35)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

0422 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

